

UNIVERSIDAD COMPLUTENSE DE MADRID
FACULTAD DE INFORMÁTICA
DEPARTAMENTO DE INGENIERÍA DEL SOFTWARE E INTELIGENCIA
ARTIFICIAL



ESTRATEGIAS DE RECOMENDACIÓN BASADAS EN CONOCIMIENTO
PARA LA LOCALIZACIÓN PERSONALIZADA DE RECURSOS EN
REPOSITORIOS EDUCATIVOS

TESIS DOCTORAL DE:
ALMUDENA RUIZ INIESTA

DIRIGIDA POR:
MERCEDES GÓMEZ ALBARRÁN
GUILLERMO JIMÉNEZ DÍAZ

Madrid, 2014

Estrategias de recomendación basadas en conocimiento para la localización personalizada de recursos en repositorios educativos



TESIS DOCTORAL

Almudena Ruiz Iniesta

Departamento de Ingeniería del Software e Inteligencia Artificial

Facultad de Informática

Universidad Complutense de Madrid

Noviembre de 2013

Documento maquetado con T_EX_S v.1.0+.

Este documento está preparado para ser imprimido a doble cara.

Estrategias de recomendación basadas en conocimiento para la localización personalizada de recursos en repositorios educativos

Memoria que presenta para optar al título de Doctor en Informática

Almudena Ruiz Iniesta

Dirigida por los Doctores

Mercedes Gómez Albarrán y Guillermo Jiménez Díaz

**Departamento de Ingeniería del Software e Inteligencia
Artificial**

**Facultad de Informática
Universidad Complutense de Madrid**

Noviembre de 2013

A mis padres

Agradecimientos

*Si quieres llegar rápido, ve solo. Si
quieres llegar lejos, ve acompañado*

Proverbio africano

Cuando uno va llegando al final de un viaje no puede evitar echar la vista atrás y ver el camino recorrido. Cuando yo hoy hago eso y miro atrás veo que el camino recorrido ha sido largo pero que nunca ha sido un camino solitario. Y cuando una reconoce en su vida la presencia de tanta gente que acompaña no puede hacer otra que dar las gracias.

El primer gracias y el más grande es para mis directores Mercedes y Guillermo, sin vosotros esta tesis no sería la misma. Todavía recuerdo aquel día en el que Guille, con esa pasión que le caracteriza, me contaba qué era eso de investigar, después de aquello no pude hacer otra cosa que lanzarme yo también a este mundo. Gracias porque desde el principio trabajasteis mucho para llevar adelante todo el trabajo. Gracias porque hoy cuando miro todo el trabajo hecho y cómo ha quedado esta tesis creo que es la mejor tesis del mundo mundial. Gracias porque en este tiempo hemos compartido más que trabajo y siempre habéis tenido un rato para mi. Gracias porque con la experiencia que dan los años y la confianza que da el tiempo compartido puedo decir que para los tres este camino ha sido una experiencia de las buenas, de esas que se llevan en la mochila y que siempre se sacan. Gracias por hacer de estos años de trabajo una experiencia así.

Por supuesto no puedo olvidar a mis compañeros del grupo de investigación GAIA que con las GAIAReuniones y los GAIASeminarios hemos aprendido los unos de los otros. Gracias porque siempre teníais alguna sugerencia para seguir avanzando y mejorando el trabajo.

Gracias a todos los compañeros del 411 que me han acompañado con la sabiduría que da la experiencia. Y por aquellas horas de experimento en las que participaron y colaboraron con sus comentarios y sugerencias. Gracias Susi por compartir el mismo camino, ¡no te queda nada!. Gracias Javi, mi maestro de la vida como a él le gusta decir, por saber cuándo regañarme y cuándo escuchar y animar. A los profesores que desde el principio estuvieron dispuestos a que pudiera realizar los experimentos que quisiera en su clases,

Pilar, Luis y Kiko.

Un agradecimiento más formal a los proyectos TIN2009-13692-C03-03 del Ministerio de Ciencia e Innovación y Proyecto IPT-2011-1890-430000 Análisis de Movimiento y Personalización Libre e Inteligente de Avatares: AMPLIA que parcialmente han financiado este trabajo.

Y ahora llega el momento de nombrar a todos aquellos que, sin saber mucho de qué iba esto, también han emprendido viaje conmigo; sin vuestras preguntas (todavía alguno no sabe muy bien qué hago) y vuestros ánimos el viaje hubiera sido distinto. Sois muchos los que en este tiempo habéis aguantado mi *pitch* de recomendadores, quizá ahora no diga tu nombre pero si estas leyendo esto sabes que tú también estás en este gracias.

David, Enrique, Pablo y Natalia, una parte de los friki-engineers que poco a poco hemos compartido las distintas tesis con sus buenos momentos, los viajes a visitar los lugares de estancias, y los no tan buenos. David, el próximo eres tú. Gracias a Susana, Natalia y Agustín, *chancleteros* que habéis sabido cuándo hablar y cuándo callar, llevando a la máxima expresión aquello que decimos, “yo estoy bien si tú estás bien”. A mis followers favoritos @jotallorente y @portugal75 (ampliado a la familia de Aranjuez) que con vuestros *pio-pio* habéis sido compañeros de camino. Un gracias muy ampliado a todas las Vedruna y *vedrunos* que a día de hoy ya no dudan sobre lo que es un recomendador. Gracias a los *buscadores* y a mis hermanos *seguidores* por vuestro aliento y confianza. Gracias Luica por tu compañía. Irene y Marina, no, no me he olvidado de vosotras, ¡cómo hacerlo!. Empezamos juntas esta aventura de estudiar 1's y 0's y vosotras sabíais antes que yo que haría una tesis. En todos estos años hemos compartido mucha vida y sin vuestros correos mañaneros (y del día en general...) la vida sería distinta. Por eso esta tesis también es un poco vuestra. Gracias por los empujones (y las collejas) que en este tiempo me habéis dado. Gracias mamá, papá, Edu porque vosotros me habéis enseñado lo fundamental. Sin vosotros y vuestro esfuerzo no habría podido estudiar la carrera. Gracias por alimentarme por dentro y por fuera todo este tiempo, por preguntar con miedo y por compartir las alegrías que la tesis nos iba regalando. Gracias por estar siempre a mi lado, sin vosotros nada sería posible. Gracias David porque desde el principio sabías que éramos tres y conseguiste que poco a poco nos convirtiéramos en uno. Gracias porque en este tiempo has sido mi entrenador, animador, mayordomo, asistente, apoyo, consuelo y un montón de cosas más. Has aguantado los no y hemos disfrutado de los sí. Sabes tan bien como yo cuál ha sido el camino para llegar hasta aquí, por eso también sabes que esto no es mio sino que es nuestro. Y como no podía ser de otra manera, Gracias a Ti que todo lo viste bueno y has confiando en mi.

Gracias a la vida que me ha dado tanto...
Noviembre de 2013

Resumen

La abundancia de recursos disponibles en repositorios educativos plantea un reto: la necesidad de proporcionar soporte a la localización de aquellos recursos que se adapten a las necesidades, objetivos, preferencias, etc. de los usuarios, en definitiva, a la localización de los recursos que les resulten más convenientes según el contexto. Además es conveniente que esta localización sea capaz de proponer listas de recursos que no contengan muchos elementos y que estos sean lo más variados posibles. Finalmente los usuarios echan en falta la existencia de mecanismos de interacción que permitan explorar el espacio de los recursos y que reduzcan el esfuerzo a realizar para localizar un recurso.

Los sistemas de recomendación, que actúan sugiriendo productos a usuarios, nacen con el propósito de facilitar la toma de decisiones en dominios y situaciones en los que las posibilidades de elección son muchas y variadas. Aunque tradicionalmente los sistemas de recomendación se han aplicado al campo del comercio electrónico, su uso se ha extendido a otros campos entre los que se encuentra el dominio educativo.

El trabajo presentado en esta memoria de tesis se engloba dentro de la línea de investigación que afronta el traslado de técnicas de recomendación al dominio educativo. En concreto, este trabajo aborda el diseño y el uso de estrategias de recomendación basadas en conocimiento como soporte al acceso personalizado a recursos educativos existentes en repositorios electrónicos. Las estrategias presentadas en este trabajo hacen uso de una representación del dominio rica en conocimiento, promueven la personalización haciendo uso de la información contextual de la actividad y del estudiante, introducen variedad en los recursos recomendados y exploran un modelo de interacción proactivo sobre el repositorio de recursos educativos que se complementa con un modelo de navegación por propuesta.

En primer lugar, en este trabajo se realizará un análisis del estado del arte en los sistemas de recomendación, extrayendo así las principales características de estos sistemas y las líneas de investigación abiertas. Después, se estudiarán los trabajos existentes en la literatura que abordan el traslado de las técnicas de recomendación al dominio educativo. Gracias a este doble análisis detectaremos las carencias de las aproximaciones propuestas y se pro-

pondrán nuevas estrategias de recomendación que alivian los inconvenientes detectados. A continuación se propondrá un framework que permite el prototipado rápido de sistemas de recomendación que siguen un modelo como el de las estrategias propuestas. Finalmente, estas estrategias serán aplicadas sobre un repositorio de recursos de Programación de Computadores y serán evaluadas desde dos puntos de vista diferentes. El primero desde un punto de vista del comportamiento de la estrategia, evaluando la calidad de las listas de recursos recomendados. El segundo será una evaluación por parte de profesores y estudiantes en un escenario real de aprendizaje-enseñanza de la Programación.

Índice

Agradecimientos	IX
-----------------	----

Resumen	XI
---------	----

I Estrategias de recomendación basadas en conocimiento para la localización personalizada de recursos en repositorios educativos	1
---	----------

1. Introducción	3
------------------------	----------

1.1. Estructura de la memoria	6
1.2. Publicaciones	8

2. Sistemas de recomendación y repositorios educativos	11
---	-----------

2.1. Técnicas de Recomendación	13
2.1.1. Recomendadores basados en filtrado colaborativo . . .	13
2.1.2. Recomendadores basados en contenido	18
2.1.3. Recomendadores basados en conocimiento	21
2.1.4. Estrategias de recomendación híbrida	25
2.2. Líneas de investigación abiertas en los sistemas de recomendación	29
2.2.1. Incorporación de diversidad en el proceso de recomendación	29
2.2.2. Estrategias de recomendación sensibles al contexto . .	35
2.2.3. Estrategias alternativas de interacción usuario recomendador	38
2.3. Sistemas de recomendación en el ámbito educativo	40
2.3.1. Repositorios de recursos educativos	41
2.3.2. Recomendadores en educación	44
2.4. Resumen y conclusiones	50

3. Estrategias de recomendación para repositorios de objetos	
---	--

de aprendizaje	53
3.1. Planteamiento y debilidades de una primera estrategia de recomendación	54
3.2. Afrontando debilidades mediante nuevas estrategias de recomendación basadas en conocimiento	57
3.3. Las fuentes de conocimiento	59
3.3.1. La ontología del dominio	60
3.3.2. Objetos de aprendizaje	60
3.3.3. Información contextual	62
3.4. Estrategia de recomendación basada en conocimiento sensible al contexto que promueve altos niveles de personalización . . .	63
3.5. Estrategia de recomendación basada en conocimiento que promueve la diversidad	67
3.6. Estrategia de recomendación basada en conocimiento que combina proactividad y navegación por propuesta	69
3.7. Resumen y conclusiones	72
4. Un framework para el desarrollo de recomendadores basados en conocimiento en el ámbito educativo	75
4.1. Frameworks y librerías para la construcción de recomendadores	76
4.2. Etapas y ejes de variabilidad en el proceso de recomendación	79
4.3. Framework para la recomendación de objetos de aprendizaje .	81
4.3.1. Clases principales del framework	81
4.3.2. Puntos flexibles del framework	85
4.3.3. Algunas implementaciones de las clases abstractas . .	90
4.4. Instanciación de las estrategias propuestas	95
4.4.1. Caso 1: Estrategia de recomendación basada en conocimiento sensible al contexto que promueve altos niveles de personalización	96
4.4.2. Caso 2: Estrategia de recomendación basada en conocimiento que promueve la diversidad	99
4.4.3. Caso 3: Estrategia de recomendación basada en conocimiento que combina proactividad y navegación por propuesta	103
4.5. Conclusiones	108
5. Evaluación de las estrategias de recomendación	109
5.1. Aplicación de las estrategias de recomendación	110
5.1.1. Los objetos de aprendizaje	111
5.1.2. La ontología del dominio	112
5.1.3. La información contextual	114
5.2. Análisis del comportamiento y evaluación computacional . . .	115

5.2.1.	Análisis del comportamiento general de las estrategias de recomendación	116
5.2.2.	Evaluación experimental de la calidad de las listas de recomendación considerando su tamaño y el orden de sus elementos	124
5.3.	Evaluando las estrategias de recomendación	134
5.3.1.	Resultados para la estrategia basada en conocimiento sensible al contexto que promueve altos niveles de personalización	141
5.3.2.	Resultados para la estrategia basada en conocimiento que promueve la diversidad	143
5.3.3.	Resultados para la estrategia basada en conocimiento que combina proactividad y navegación por propuesta	147
5.4.	Conclusiones	148
6.	Conclusiones y trabajo futuro	153
6.1.	Conclusiones	154
6.2.	Trabajo futuro	160
 II Summary: Knowledge-based recommendation strategies for personalized access to learning object repositories		
7.	Introduction	165
7.1.	Outline	168
7.2.	Publications	169
8.	Knowledge-based recommendation strategies for learning object repositories	171
8.1.	Knowledge-based recommendation strategies for recommending learning objects	173
8.2.	The knowledge sources	174
8.2.1.	The learning objects	174
8.2.2.	The domain ontology	174
8.2.3.	The contextual information	175
8.3.	A knowledge-based context-aware recommendation strategy that promotes high levels of personalization	175
8.4.	A knowledge-based recommendation strategy that promotes diversity	178
8.5.	A proactive knowledge-based recommendation strategy that exploits navigation-by-proposing	180
8.6.	Conclusions	181

9. A framework for rapid prototyping of knowledge-based recommender systems in the learning domain	183
9.1. Stages of the recommendation process	184
9.2. A framework for learning object recommendation	186
9.2.1. The main classes of the framework	186
9.2.2. The framework hooks	189
9.2.3. The implementation of some concrete classes	191
9.3. Conclusions	193
10. Evaluation of the knowledge-based recommendation strategies	195
10.1. Applying the approach to an educational repository of programming learning objects	196
10.2. Evaluation of the strategies	197
10.2.1. Experimental analysis of the behaviour	198
10.2.2. Evaluation with users	201
10.2.3. Conclusions	204
11. Conclusions and future work	205
11.1. Conclusions	205
11.2. Future work	208
Bibliografía	211

Índice de figuras

3.1. Esquema de la aproximación del modelo de gestión de repositorios de objetos de aprendizaje (se usan las siglas LO –del inglés <i>Learning Object</i> – para referirse a los recursos educativos).	59
3.2. Diagrama de interacción usuario-recomendador para la estrategia proactiva.	70
4.1. Diseño de clases que conforman el framework propuesto	84
4.2. Paso de mensajes producidos por la ejecución del método <code>init</code> de la clase <code>KBRecommender</code> .	88
4.3. Paso de mensajes producidos por la ejecución del método <code>initRecommendation</code> de la clase <code>KBRecommender</code> .	89
4.4. Paso de mensajes producidos por la ejecución del método <code>recommend</code> de la clase <code>KBRecommender</code> .	89
4.5. Paso de mensajes producidos por la ejecución del método <code>endRecommendation</code> de la clase <code>KBRecommender</code> .	90
4.6. Diseño UML de las etapas de obtención de consulta proporcionadas en el framework.	92
4.7. Diseño UML de las etapas de recuperación proporcionadas en el framework.	92
4.8. Diseño UML del filtro proporcionado en el framework	93
4.9. Diseño UML de las métricas de calidad que componen el framework	94
4.10. Diseño UML de las etapas de selección proporcionadas en el framework	95
4.11. Diagrama de objetos de la estrategia de recomendación sensible al contexto que promueve altos niveles de personalización	98
4.12. Diagrama de objetos de la estrategia que promueve la diversidad en el conjunto de recomendados	102
4.13. Paso de mensajes producido por la ejecución de una recomendación en <i>Navigation</i>	105
4.14. Diagrama de objetos de la estrategia que utiliza un modelo de navegación por propuesta	107

5.1. Vista parcial de la ontología del dominio	113
5.2. Resumen de resultados sobre las relevancias parciales en la lista de objetos recomendados utilizando las métricas de calidad C_{1p} (Ecuación (5.6)) y C_{2p} (Ecuación (5.7)) en <i>KBpersonalization</i>	120
5.3. Resumen de resultados sobre las relevancias parciales en la lista de objetos recomendados utilizando las métricas de calidad C_{1d} (Ecuación (5.8)) y C_{2d} (Ecuación (5.9)) en <i>KBdiversity</i>	123
5.4. Comparación de $NDCG_{Sim}$ y $NDCG_{UP}$ en función de α y k en <i>KBpersonalization</i> con C_{1p} (Ecuación 5.6) como métrica de calidad	128
5.5. Función de compromiso para <i>KBpersonalization</i> con C_{1p} (Ecuación 5.6).	129
5.6. Comparación de $NDCG_{Sim}$ y $NDCG_{UP}$ en función α y k en <i>KBpersonalization</i> con C_{2p} (Ecuación 5.7) como métrica de calidad.	130
5.7. Función de compromiso para <i>KBpersonalization</i> con C_{2p} (Ecuación 5.7).	131
5.8. Comparación de $NDCG_{Sim}$ y $NDCG_{Div}$ en función de α y k en <i>KBdiversity</i> con C_{1d} (Ecuación (5.8)) como métrica de calidad.	132
5.9. Función de compromiso para <i>KBdiversity</i> con C_{1d} (Ecuación 5.8).	133
5.10. Comparación de $NDCG_{Sim}$ y $NDCG_{Div}$ con α y k en <i>KBdiversity</i> con C_{2d} (Ecuación (5.9)) como métrica de calidad.	135
5.11. Función de compromiso para <i>KBdiversity</i> con C_{2d} (Ecuación (5.9)).	136
5.12. Interfaz de <i>KBPersonalization</i> y <i>KBdiversity</i>	138
5.13. Interfaz de <i>KBnavigation</i> (arriba estado inicial, abajo resultado de una recomendación)	139
5.14. Frecuencias acumuladas para las valoraciones sobre el rendimiento (profesores a la izquierda, estudiantes a la derecha) para <i>KBpersonalization</i>	142
5.15. Frecuencias acumuladas para las valoraciones sobre el esfuerzo (profesores a la izquierda, estudiantes a la derecha) para <i>KBpersonalization</i>	143
5.16. Frecuencias acumuladas para las valoraciones sobre la intención de uso (profesores a la izquierda, estudiantes a la derecha) para <i>KBpersonalization</i>	144
5.17. Frecuencias acumuladas para las valoraciones sobre el rendimiento (profesores a la izquierda, estudiantes a la derecha) para <i>KBdiversity</i>	145

5.18. Frecuencias acumuladas para las valoraciones sobre el esfuerzo (profesores a la izquierda, estudiantes a la derecha) para <i>KBdiversity</i>	146
5.19. Frecuencias acumuladas para las valoraciones sobre la intención de uso (profesores a la izquierda, estudiantes a la derecha) para <i>KBdiversity</i>	146
5.20. Frecuencias acumuladas para las valoraciones sobre el rendimiento (profesores a la izquierda, estudiantes a la derecha) para <i>KBnavigation</i>	148
5.21. Frecuencias acumuladas para las valoraciones sobre el esfuerzo (profesores a la izquierda, estudiantes a la derecha) para <i>KBnavigation</i>	149
5.22. Frecuencias acumuladas para las valoraciones sobre la intención de uso (profesores a la izquierda, estudiantes a la derecha) para <i>KBnavigation</i>	149

Índice de Tablas

5.1. Pérdida de similitud media con la consulta y ganancia de utilidad pedagógica media en base a la aproximación de referencia basada en similitud pura ($\alpha = 1$).	121
5.2. Pérdida de similitud media con la consulta y ganancia de diversidad en base a la aproximación de referencia basada en similitud pura ($\alpha = 1$).	124
5.3. Preguntas relacionadas a cada uno de los objetivos de la evaluación	137

Parte I

Estrategias de recomendación basadas en conocimiento para la localización personalizada de recursos en repositorios educativos

Capítulo 1

Introducción

*Todo viaje, por largo que sea, empieza
por un solo paso.*

Lao Tse

Diariamente estamos expuestos a una cantidad de información que aumenta mucho más rápido que nuestra capacidad para procesarla. Esto también sucede en el ámbito educativo, donde en los últimos años, y en la mayoría de las disciplinas educativas, existe una tendencia por parte de los docentes a desarrollar contenidos educativos que son puestos a disposición de los estudiantes. Los repositorios de recursos electrónicos resultantes pretenden así facilitar y fomentar el proceso de (auto)aprendizaje.

Sin embargo, la abundancia de recursos educativos en estos repositorios es un arma de doble filo y plantea un reto adicional: es necesario proporcionar soporte a la localización de aquellos recursos que se adapten a las necesidades, objetivos, preferencias, etc. de los estudiantes, en definitiva, a la localización de los recursos que les resulten más convenientes según el contexto. Los recursos localizados deben ser adecuados al estudiante para que éste pueda aprovechar al máximo una sesión de estudio. Esto significa que las propuestas no pueden contener un número excesivo de recursos y además sería deseable que éstas fueran lo más variadas posibles, evitando que el estudiante obtenga siempre recursos muy similares entre sí. Finalmente cabe señalar que los usuarios de los repositorios muestran también interés en disponer de mecanismos de interacción que permitan navegar por el espacio de los recursos y que alivien el coste de comunicación de los requisitos del usuario. Pasar por alto estas necesidades puede desembocar en la infrautilización de los citados repositorios y, en consecuencia, en no conseguir el efecto positivo deseado de cara al proceso de aprendizaje, a pesar del gran esfuerzo que supone desarrollar todos los contenidos educativos allí recogidos.

Desde mediados de los años 90 el trabajo de investigación en técnicas de recomendación ha tenido como objetivo precisamente aliviar la carga que

supone la preselección de información, productos, recursos, etc. en los que los usuarios de los mismos pudiesen estar interesados. Así pues, los sistemas de recomendación nacen con el propósito de facilitar la toma de decisiones en dominios y situaciones en los que las posibilidades de elección son muchas y variadas. Actúan sugiriéndonos buenos productos y/o servicios bien sea para comprar algo o para consumir. Aunque tradicionalmente los sistemas de recomendación se han aplicado al campo del comercio electrónico, su uso se ha trasladado a muchos y diversos campos entre los que también se incluye el campo académico. En este sentido, existen algunas investigaciones que han llevado a cabo el desarrollo de herramientas de recomendación que sugieren a los estudiantes aquellos recursos educativos que pudieran ser de su interés.

El trabajo presentado en esta memoria de tesis está englobado dentro de esta línea de investigación que afronta el traslado de técnicas de recomendación al ámbito del e-learning. De acuerdo a esta idea, este trabajo de tesis presenta una aproximación a la utilización de los sistemas de recomendación como soporte al acceso personalizado a recursos educativos existentes en repositorios electrónicos. Este trabajo ha conducido a la definición de tres estrategias de recomendación que hacen uso del conocimiento existente del dominio de aplicación, así como de información adicional tanto del estudiante como de la actividad, con una representación semántica basada en ontologías. Con el desarrollo de estas estrategias de recomendación hemos explorado, el proporcionar, a un estudiante dado, un conjunto de recursos educativos que se adaptan a sus necesidades de aprendizaje, el promover la variedad de recursos en el conjunto recomendado y el explorar un modelo de interacción usuario-recomendador que combina la proactividad con la navegación por propuesta. La implementación de estas tres estrategias ha permitido detectar similitudes y diferencias entre ellas y ha conducido a la propuesta de un framework que permite el prototipado rápido de recomendadores basados en conocimiento para el ámbito del e-learning. Para estas tres estrategias se han desarrollado sus respectivos prototipos que han sido evaluados tanto de una manera computacional, como por profesores y estudiantes para determinar su satisfacción con las estrategias diseñadas.

Estas tres estrategias presentadas surgen al querer abordar las carencias detectadas en los sistemas de recomendación en el ámbito educativo. En los trabajos que se ocupan del traslado de las técnicas de recomendación al ámbito educativo hemos detectado que no aprovechan del todo las características propias que tiene dicho dominio. En particular, hemos visto que son pocos los trabajos que sacan partido al conocimiento sobre el dominio y sobre el estudiante (objetivos de aprendizaje, estilos de aprendizaje, nivel de conocimiento del estudiante, etc.), reduciendo las técnicas de recomendación utilizadas a aquellas que recomiendan recursos que son populares entre los estudiantes. Utilizar una representación rica del conocimiento puede repercutir en obtener recomendaciones mejor adaptadas al estudiante y

sus necesidades y aumentar la satisfacción con el recomendador. De ahí que las tres estrategias presentadas en este trabajo de tesis pertenezcan a los sistemas de recomendación basados en conocimiento.

Además no podemos olvidar que el conjunto de recursos recomendados deben servir para que el estudiante aproveche al máximo una sesión de estudio. Esto significa que las listas de recomendación propuestas no deben estar sobreespecializadas, es decir, no conviene que contengan recursos muy similares entre sí. Esto podría ocasionar un desinterés por parte del estudiante con el uso del recomendador. Los trabajos sobre recomendación en e-learning parecen no afrontar este problema.

La última carencia detectada en los trabajos que se ocupan del traslado de las técnicas de recomendación al e-learning es la falta de mecanismos de interacción usuario-recomendador que aligeren la carga de trabajo al estudiante. Es decir, se echa en falta utilizar mecanismos de interacción más proactivos y con mecanismos de interacción navegacionales, que guíen al estudiante por el espacio de recursos sin que éste tenga que realizar un gran esfuerzo para encontrar aquellos recursos que mejor se adaptan a sus necesidades.

Una vez expuestas las principales ideas del trabajo de tesis, a continuación se enumeran los principales objetivos de la misma:

- Realización de un análisis exhaustivo del estado del arte en los sistemas de recomendación. Este análisis nos permitirá establecer aspectos de diseño claves en los sistemas de recomendación y cuáles son las principales características de estos sistemas, sus beneficios y algunos de los principales problemas que nos podemos encontrar en ellos.
- Análisis de algunas líneas de investigación abiertas en los sistemas de recomendación, (a) inclusión de diversidad en las listas de recomendación, (b) incorporación de la información contextual para mejorar la personalización de las recomendaciones y (c) estudio de modelos de interacción usuario-recomendador que alivien el esfuerzo que supone para un usuario enfrentarse a la realización de una consulta al sistema.
- Realización de un análisis sobre el traslado de los sistemas de recomendación al ámbito del e-learning y detección de los tipos de conocimiento sobre los que un recomendador puede sacar partido, como el estado cognitivo del estudiante o las estrategias y caminos de aprendizaje que resulten efectivos en un contexto educativo concreto.
- Diseño de estrategias genéricas de recomendación que hagan uso de una representación rica en conocimiento y que además promuevan altos niveles de personalización, incluyan diversidad en las propuestas y/o exploren un modelo de interacción que requiera poco esfuerzo por parte del usuario.

- Reconocimiento de las necesidades de conocimiento específicas requeridas para el diseño de las estrategias así como el uso de técnicas que permitan la representación del conocimiento necesario.
- Desarrollo de un framework orientado a objetos para el prototipado rápido de recomendadores basados en conocimiento aplicados al dominio del e-learning.
- Implementación de prototipos que instancien distintas variantes de cada estrategia propuesta con el fin de determinar qué instanciación obtiene mejores resultados.
- Aplicación de las estrategias genéricas a repositorios en un dominio de aplicación concreto como es la enseñanza de la Programación.
- Análisis del comportamiento de las estrategias de recomendación resultantes en lo que se refiere a la calidad de las listas de recomendación propuestas.
- Evaluación de la aplicación de las estrategias diseñadas en una comunidad real de profesores y estudiantes, que pueda dar información sobre la satisfacción con las recomendaciones recibidas y arrojar luz sobre posibles mejoras.

1.1. Estructura de la memoria

El presente trabajo está estructurado en seis capítulos, siendo el primero de ellos esta introducción. A continuación se muestra la secuencia del resto de capítulos, así como una breve descripción de su contenido:

Capítulo 2. Sistemas de recomendación y repositorios educativos.

Este capítulo introduce las principales áreas de investigación relacionadas con este trabajo: sistemas de recomendación y repositorios de recursos educativos. Para ambas áreas se hará un análisis del estado del arte identificando cuáles son las principales características y las líneas de investigación abiertas. En cuanto a los sistemas de recomendación, junto con la revisión del estado actual de la investigación se mostrarán algunos ejemplos de sistemas que, a nuestro juicio, reúnen las principales características. Se expondrán junto con estos sistemas algunas de las limitaciones propias de los sistemas de recomendación. Además se presentarán tres líneas de investigación abiertas que abordan sendos problemas detectados en los sistemas de recomendación. En lo que respecta a los repositorios de recursos educativos, analizaremos trabajos que presentan adaptaciones de las técnicas de recomendación a la localización de recursos. Asimismo, esta revisión nos permitirá

analizar las carencias existentes en el área y en las que este trabajo de tesis supone una aportación.

Capítulo 3. Estrategias de recomendación para repositorios de objetos de aprendizaje. Comenzaremos este capítulo describiendo la estrategia de recomendación de objetos de aprendizaje que ha servido de punto de partida al presente trabajo de tesis y expondremos los inconvenientes achacables a la misma: niveles de personalización mejorables, inclusión de diversidad no contemplada y carencia de alternativas navegacionales proactivas de interacción. Atendiendo a las líneas de mejora detectadas, se presentarán tres nuevas estrategias que hacen uso de una representación rica en conocimiento y ayudan a solventar los inconvenientes de la inicial. La descripción de las nuevas estrategias se hará de manera genérica, incluyendo tanto el conocimiento necesario (ontología del dominio, información contextual y representación de los objetos de aprendizaje) como su funcionamiento.

Capítulo 4. Un framework para el desarrollo de recomendadores basados en conocimiento en el ámbito educativo. Este capítulo describe el framework desarrollado para el prototipado rápido de sistemas de recomendación basados en conocimiento aplicados a la recomendación de objetos de aprendizaje. La principal característica del framework es que parte de un esquema de recomendación orientado a cinco etapas que pueden ser configurables y adaptadas a cada recomendador concreto. El framework no sólo proporciona implementaciones de enfoques alternativos para cada una de las etapas, sino que ha sido diseñado de modo que se pueda extender fácilmente con la inclusión de nuevas implementaciones. Finalmente se ejemplifica el uso del framework con la implementación de tres recomendadores distintos.

Capítulo 5. Evaluación de las estrategias de recomendación. Este capítulo presenta una evaluación de las estrategias de recomendación propuestas. Se realiza, en primer lugar, un análisis del comportamiento de las estrategias atendiendo a tres características de las recomendaciones producidas y a la calidad de las mismas. En segundo lugar se describen los resultados de un experimento llevado a cabo con profesores y alumnos de la Facultad de Informática de la Universidad Complutense de Madrid, en el que se evalúa la satisfacción obtenida por dichos usuarios al utilizar sistemas de recomendación sobre un repositorio de recursos de Programación de Computadores. En este experimento dichos usuarios utilizaron distintos prototipos de las estrategias de recomendación diseñadas en esta tesis.

Capítulo 6. Conclusiones y trabajo futuro. En este capítulo se traza un breve recorrido por lo que ha supuesto desarrollar este trabajo de

tesis y presenta las principales aportaciones que se han producido con su elaboración. Para terminar, se presentan las que, a criterio de la autora, constituyen las principales líneas de trabajo futuro que pueden seguirse desde el punto en el que se ha finalizado el trabajo.

1.2. Publicaciones

El trabajo descrito en esta memoria de tesis ha generado una serie de publicaciones a lo largo de los últimos años. Aunque cada una de ellas ha sido referenciada en esta memoria en el lugar en el que se trata su contenido, aquí se enumeran, a modo de resumen:

1. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Recommendation in repositories of learning objects: A proactive approach that exploits diversity and navigation-by-proposing. En *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies*, páginas 543–545. IEEE Computer Society, 2009. ISBN 978-0-7695-3711-5.
2. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. User-adaptive recommendation techniques in repositories of learning objects: Combining long-term and short-term learning goals. En *Proceedings of the 4th European Conference on Technology Enhanced Learning: Learning in the Synergy of Multiple Disciplines*, páginas 645–650. Springer, 2009. ISBN 978-3-642-04635-3.
3. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Promoting strong personalization in content-based recommendation systems of learning objects. En *Proceedings of the XI International Symposium on Computers in Education*. 2009. ISBN 978-989-20-1774-7.
4. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Personalización en recomendadores basados en contenido y su aplicación a repositorios de objetos de aprendizaje. *IEEE-RITA*, vol. 5(1), páginas 31–38, 2010. ISSN 1932-8540.
5. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Combining personalization and diversity in a case-based recommendation strategy for the learning domain. En *Proceedings of the International Council for Educational Media and International Symposium on Computers in Education Joint Conference*, páginas 409–419. 2011. ISBN 978-972-789-347-8.
6. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. An experimental analysis of the behaviour of a personalized case-based

- recommendation strategy for the learning domain. En *Proceedings of the 19th International Conference on Computers in Education*, páginas 135–137. National Electronics and Computer Technology Center, 2011. ISBN 978-616-12-0188-3.
7. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Un framework para el desarrollo de recomendadores basados en contenido aplicados a objetos de aprendizaje. En *Actas del 2º Taller sobre Ingeniería del Software en eLearning*, páginas 147–161. Universidad Complutense de Madrid, Área de Ciencias Exactas y de la Naturaleza, 2011. ISBN 978-84-694-7325-2.
 8. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. A framework for rapid prototyping of knowledge-based recommender systems in the learning domain. *Journal of Research and Practice in Information Technology*, vol. 44(2), páginas 167–181, 2012. ISSN 1443-458X.
 9. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. A hybrid user-centered recommendation strategy applied to repositories of learning objects. *International Journal of Web Based Communities*, vol. 8(3), páginas 302–321, 2012. ISSN 1741-8216.

Capítulo 2

Sistemas de recomendación y repositorios educativos

Estamos dejando atrás la era de la información para entrar en la era de la recomendación.

Chris Anderson

Con el desarrollo de las tecnologías de la información ha habido una explosión de la información a la que una persona puede acceder en cualquier momento sobre cualquier tema. Una muestra de esto es la afirmación publicada en noviembre de 2006 por el diario *The economist*: “*People read around 10 MB worth of material day, hear 400MB a day, and see 1 MB of information every second*”. En otras palabras, estamos sobrecargados de información haciendo difícil la tarea de discernir cuál es la que más nos interesa o la que mejor se adapta a nuestras necesidades. Para manejar toda esta información con la que nos encontramos cada día, necesitamos ayuda.

Toda esta cantidad de información que tenemos a nuestro alcance nos lleva a que cada vez que compramos algo, decidimos qué libro leer, qué película ver o dónde ir de vacaciones, nos encontramos con una gran variedad de alternativas. Tradicionalmente la manera de decidir qué alternativa era mejor era recurrir a la información asociada al producto, o bien recurrir a amigos, conocidos o expertos en el área que nos ayudaban a tomar decisiones sobre qué producto podía ser el mejor para nosotros. Pero ahora, en la era de la información, recurrimos a la web –a los buscadores– para encontrar respuestas a nuestras preguntas. El problema que nos encontramos en la web es que para encontrar una buena respuesta tenemos que realizar bien la pregunta, y en muchas ocasiones no sabemos formular bien dicha pregunta –generalmente porque nos faltan conocimientos del área. Pero también, en otras ocasiones, es el buscador el que no sabe proporcionarnos una buena respuesta adaptada a nosotros porque no sabe cuáles son nuestros gustos,

nuestros intereses y/o necesidades en este momento o porque no tiene un conocimiento sobre el producto o dominio. Por todo esto son necesarios nuevos mecanismos que nos ayuden a localizar aquellos productos que sean los mejores para nosotros. Y con este propósito nacieron los recomendadores.

La definición más sencilla que podemos dar sobre lo que es un recomendador es la dada por Jannach et al. (2010): *Los recomendadores son sistemas que ayudan a emparejar usuarios con productos*. Es decir, a partir de un conjunto de productos muy amplio en el que un usuario puede tomar distintas decisiones sobre qué producto escoger, el recomendador le ayuda en esta tarea seleccionando para él un subconjunto de todos los productos que mejor se adaptan a sus necesidades.

Los recomendadores emergieron como un área de investigación independiente a mediados de los 90 y tradicionalmente han sido aplicados en el campo del comercio electrónico (por ejemplo para recomendar productos que comprar en tiendas on-line, películas que ver o música que escuchar). Pero su uso se ha ido extendiendo y ahora podemos encontrar recomendadores sobre casi cualquier cosa, por ejemplo personas a las que seguir en redes sociales, lugares que visitar o aplicaciones móviles que descargar. Su uso también se ha trasladado al ámbito educativo. En este sentido, algunas de las investigaciones se han traducido en el desarrollo de herramientas de recomendación para sugerir cursos y actividades curriculares, dar soporte personalizado en sistemas de gestión de contenido o sugerir recursos educativos existentes en repositorios.

Este trabajo de tesis doctoral se enmarca dentro del campo de los sistemas de recomendación, y más concretamente en la aplicación de estos sistemas al ámbito educativo. En el presente capítulo presentamos una revisión del estado del arte sobre los sistemas de recomendación. Comenzaremos proporcionando un marco donde clasificar los tipos que existen y presentando técnicas y sistemas recomendadores de referencia (Sección 2.1). Completaremos la revisión de los sistemas de recomendación introduciendo tres líneas de investigación abiertas en las que este trabajo de investigación se apoya (Sección 2.2): la incorporación de diversidad en los resultados de recomendación mostrados, la explotación de la información contextual en estos sistemas y la exploración de modelos de interacción alternativos a la formulación de una consulta directa que alivien la carga de trabajo del usuario. Una vez presentado el campo de los sistemas de recomendación pasaremos a explicar cuál está siendo el proceso de incorporación de estos sistemas al dominio de la educación (Sección 2.3). En un primer momento presentaremos los repositorios de contenidos educativos, junto con las características que los definen y algunos ejemplos (Sección 2.3.1). Posteriormente se realizará un estudio sobre lo que se está realizando en sistemas de recomendación para el ámbito educativo en cuanto a los repositorios se refiere (Sección 2.3.2). Terminaremos el capítulo con un pequeño resumen de las principales características

vistas así como algunas conclusiones extraídas después de la revisión de los sistemas (Sección 2.4).

2.1. Técnicas de Recomendación

Generalmente un sistema de recomendación se basa, para realizar las recomendaciones, en el uso de la información disponible sobre las preferencias de los usuarios, de su interacción con el recomendador y/o de los productos. En base a esta información existen tres grandes familias de recomendadores (Jannach et al., 2010): los basados en filtrado colaborativo (Schafer et al., 2007), que son aquellos que utilizan valoraciones asociadas a los productos dadas por el propio usuario y/o por otros usuarios; los basados en contenido (Pazzani y Billsus, 2007), que son aquellos que realizan la recomendación al usuario en base a la descripción de los productos; y los basados en conocimiento (Burke, 2000), que son aquellos que utilizan toda la información disponible sobre el usuario, sobre los productos y sobre el dominio de recomendación para efectuar las recomendaciones. Con el fin de abordar y compensar las deficiencias particulares de cada una de las estrategias anteriores, la combinación de diferentes estrategias de recomendación es una técnica habitual que da lugar a los sistemas de recomendación híbridos (Burke, 2002a).

2.1.1. Recomendadores basados en filtrado colaborativo

La primera de las técnicas que veremos, y probablemente la técnica más empleada en los sistemas recomendadores propuestos en la literatura, es la conocida como filtrado colaborativo. Un recomendador basado en filtrado colaborativo entiende que si dos usuarios han compartido algunos de sus intereses en el pasado tendrán gustos similares en el futuro. Por ejemplo, en un escenario de recomendación de películas: si dos usuarios A y B tienen una historia de gustos cinematográficos similares y A adquiere una película que B no ha visto, entonces es razonable recomendar a B esa película también.

El filtrado colaborativo empareja usuarios con preferencias similares, o bien empareja productos con similares patrones de valoración por parte de los usuarios, con el fin de realizar recomendaciones (Adomavicius y Tuzhilin, 2005; Desrosiers y Karypis, 2011). Las técnicas de filtrado colaborativo buscan predecir la utilidad de un producto para un usuario determinado de acuerdo a las valoraciones que han dado para dicho producto otros usuarios con preferencias similares o de acuerdo a las valoraciones que para productos similares ha dado dicho usuario. Estas técnicas tienen la propiedad de que no hay descripciones de los productos, simplemente aprovechan la información acerca de las valoraciones anteriores. La suposición fundamental de este tipo de recomendadores es que las valoraciones asignadas de forma implícita o

explícita a los productos disponibles, se mantienen estables y consistentes en el tiempo. De esta manera el recomendador entiende las preferencias de los usuarios y puede generar recomendaciones.

Una de las clasificaciones para las estrategias de filtrado colaborativo es la propuesta por Koren y Bell (2011) en la que se distingue entre estrategias basadas en vecinos y las basadas en modelos. Las estrategias basadas en vecinos se centran en las relaciones entre los productos o las relaciones entre los usuarios referidas anteriormente. Las estrategias basadas en modelos comprenden un enfoque alternativo mediante la transformación de usuarios y productos al mismo espacio de factores latentes. El espacio latente trata de explicar las valoraciones mediante la caracterización de usuarios y productos en factores automáticamente inferidos a partir de las respuestas del usuario. A continuación presentamos estas dos aproximaciones con algo más de detalle.

2.1.1.1. Estrategias basadas en vecinos

Las estrategias basadas en vecinos fueron las primeras en ser utilizadas y definidas, siendo las que han gozado de más popularidad en el diseño de estos recomendadores. Estas estrategias realizan la predicción de las valoraciones utilizando la colección entera de valoraciones disponibles. Dentro de las estrategias de vecinos encontramos la clasificación propuesta por Schafer et al. (2007) que distingue dos tipos: basados en usuario (*user-based*) y basados en productos (*item-based*). En ambas técnicas el objetivo es predecir la valoración que un usuario otorgaría a un determinado producto objetivo y, en función de dicha valoración, sugerirlo (o no) al usuario.

El filtrado colaborativo basado en usuario construye las predicciones sobre la valoración de un producto P para un usuario U a partir de las valoraciones que le han dado a P los vecinos de U . Para formar este grupo de usuarios vecinos la estrategia considera que dos usuarios son similares si han valorado los mismos productos en sus perfiles y les han asignado valoraciones parecidas (Herlocker et al., 2002). Los aspectos críticos de estos tipo de algoritmos son, a) cómo identificar a los vecinos que constituyen el mejor vecindario para generar las recomendaciones, y b) cómo utilizar la información proporcionada por estos vecinos.

El primer sistema en proponer y utilizar un algoritmo de filtrado colaborativo basado en usuario fue el propuesto por GroupLens (Konstan et al., 1997). En esta primera aproximación todos los usuarios del sistema eran utilizados para calcular la predicción. Los siguientes algoritmos en esta línea mejoraron la precisión y la eficiencia mediante la limitación a los k vecinos más cercanos de un usuario y usaron sólo éstos para calcular las predicciones (Herlocker et al., 1999).

Típicamente la manera de seleccionar el vecindario se basa en seleccionar aquellos usuarios que son más similares al usuario actual de acuerdo a

métricas de similitud (Desrosiers y Karypis, 2011). Una vez determinado el conjunto de productos que dos usuarios han valorado, se examina en qué medida ambos usuarios muestran comportamientos similares (por ejemplo valoraciones, navegación y patrones de elección de productos) en esos productos. Para este tipo de algoritmos se ha demostrado empíricamente que la métrica que obtiene mejores resultados en el cálculo de la similitud entre usuarios es el coeficiente de correlación de Pearson (Herlocker et al., 1999). Fijar un número máximo de vecinos (o un valor mínimo de similitud) y restringir el tamaño del vecindario, tal y como se ha indicado anteriormente, sirve para mejorar la eficiencia computacional o para evitar aquellos usuarios que no son suficientemente similares al usuario actual. Una vez que los vecinos del usuario actual han sido seleccionados, cuanto más similar sea un vecino al usuario actual, más se tienen en cuenta sus valoraciones para producir la recomendación. Un enfoque común consiste en predecir la valoración de un producto para el usuario actual mediante una combinación lineal de las valoraciones de sus vecinos, ponderado por la similitud entre el usuario actual y los vecinos de éste. Por otra parte, también existen otros trabajos que proponen métricas de similitud alternativas y modificaciones sobre cómo seleccionar y ponderar los vecinos, ya sea modificando la métrica de similitud (McLaughlin y Herlocker, 2004; Ma et al., 2007), utilizando métodos de *cluster* para calcular el vecindario (Xue et al., 2005), o por el aprendizaje de los mejores pesos para la clasificación de la predicción (Bell y Koren, 2007; Koren, 2008).

En cuanto al filtrado colaborativo basado en producto, éste construye las predicciones sobre la valoración de un producto P para un usuario U a partir de las valoraciones que ha dado U a los vecinos de P . En este tipo de sistemas, se considera que dos productos son similares (o vecinos) si los usuarios que han valorado uno de ellos tienden a valorar el otro asignándole valoraciones parecidas (Sarwar et al., 2001; Shardanand y Maes, 1995). Una ventaja de este tipo de algoritmo es que las similitudes entre productos pueden estar precalculadas, haciendo el proceso de recomendación más rápido. En este caso la métrica más ampliamente aceptada es la *similitud basada en el coseno* (Sarwar et al., 2001). Esta métrica mide la similitud entre los dos vectores de valoraciones de productos basándose en el ángulo que forman ambos. También existen otras métricas de similitud basadas en probabilidad (Deshpande y Karypis, 2004). Como en el caso de la aproximación basada en usuario, el tamaño del vecindario se limita a un número específico, es decir, no todos los vecinos son tenidos en cuenta para el cálculo de las predicciones. Después de tener construido el vecindario el siguiente paso es calcular las predicciones, siendo las métricas para realizar este cálculo las mismas que en el enfoque basado en usuario.

Tapestry fue el primer sistema que incorporó las opiniones de los usuarios a una base de datos de mensajes y a un buscador (Goldberg et al., 1992). Es-

te sistema recomendaba listas de noticias, boletines electrónicos de artículos a usuarios de correo electrónico, siguiendo un enfoque basado en producto. La tienda online Amazon.com dispone también de uno de los más conocidos y utilizados recomendadores colaborativos en el ámbito del comercio electrónico, utilizando una aproximación basada en productos (Linden et al., 2003). Cada producto adquirido y valorado por un usuario lo empareja con productos similares, y después combina aquellos productos similares en una lista de recomendación.

2.1.1.2. Estrategias basadas en modelo

Como ya dijimos, las estrategias basadas en vecinos han sido las más ampliamente utilizadas en el diseño de recomendadores colaborativos, pero éstas tienen un problema que dificulta su uso en sistemas de recomendación reales: su complejidad de cálculo es muy alta. En el caso de los sistemas basados en usuarios la complejidad es de $O(N^2 \times M \times K)$ siendo N el número de usuarios, M el número de productos y K el número de usuarios que forman parte del vecindario de cada usuario. El caso de los sistemas basados en productos es similar, la complejidad es de $O(M^2 \times N \times K)$, siendo K en este caso el número de productos vecinos. Ambos métodos tienen una complejidad cuadrática, lo que no es admisible para sistemas reales con una gran cantidad de datos. Para intentar solventar este problema surgieron otras técnicas que apuestan por un preprocesamiento offline de los datos disponibles y facilitar así el cálculo de las predicciones (Breese et al., 1998). Estas nuevas técnicas son conocidas como los recomendadores basados en modelos.

Las aproximaciones basadas en modelos construyen modelos estadísticos de patrones de valoración de usuarios/productos con el fin de obtener predictores de valoraciones automáticos. Algunas aproximaciones aprenden dichos modelos mediante algún tipo de reducción sobre la dimensionalidad con el fin de descubrir cuáles son los factores latentes entre usuarios y productos. Una de estas técnicas ampliamente adoptadas en los últimos años es el modelo propuesto por Koren et al. (2009). El éxito y popularidad alcanzado por esta propuesta es debido, principalmente, a las mejoras de precisión que ha mostrado. Este modelo se conoce como *modelo de los factores latentes* y utiliza técnicas avanzadas de factorización matricial. La factorización matricial intenta caracterizar usuarios y productos con un conjunto pequeño de factores inferidos a partir de los patrones de valoración del usuario. Por ejemplo, en el dominio de la recomendación de películas el conjunto de factores podría ser: comedia familiar, drama, miedo. Lo que hace el modelo es medir en qué grado un determinado producto cumple esos factores (de manera positiva y negativa). En Jamali y Ester (2010) utilizan este modelo para construir un recomendador colaborativo que recomienda películas –a partir del dataset de Flixter.com– haciendo uso de las redes sociales para propagar la confianza entre usuarios. En su trabajo consiguen mejorar la precisión de la técnica

gracias a la inclusión de la confianza entre usuarios. Otro trabajo que también utiliza esta técnica es el presentado por Koren y Sill (2011) en el que proponen representar las valoraciones de los usuarios como ordinales en lugar de como valores numéricos. En (Forbes y Zhu, 2011) presentan un recomendador de recetas de cocina que utiliza la técnica de la factorización matricial en el que introducen de manera manual información sobre restricciones en el modelo (adaptadas al dominio de aplicación).

Otros ejemplos son los modelos de semántica latente (Hofmann, 2004), redes neuronales (Salakhutdinov et al., 2007) o *Latent Dirichlet Allocation* (Blei et al., 2003). Entre estas estrategias también tenemos aquellas aproximaciones que utilizan modelos probabilísticos donde la predicción sobre la valoración que un usuario otorgaría a un elemento se calcula en base a distribuciones de probabilidad (Wang et al., 2006, 2008).

Para finalizar, comentaremos algunas debilidades de las técnicas de recomendación basadas en filtrado colaborativo:

- **Dispersión:** El conjunto de valoraciones usuario-producto suele ser menor que el conjunto total de usuarios-productos. Ante tal desigualdad, es menos probable que dos perfiles contengan exactamente las mismas valoraciones y, por consiguiente, es más difícil encontrar vecinos tanto para el usuario activo como para el producto objetivo, fase crucial en los enfoques colaborativos basados en usuario y en producto, respectivamente.
- **Usuarios poco habituales:** Este problema es conocido como el efecto *gray sheep*. Está relacionado con aquellos usuarios que tienen opiniones poco habituales sobre los productos disponibles. Por lo tanto, no pueden recibir buenas recomendaciones colaborativas debido a la dificultad para crear un vecindario fiable.
- **Nuevo producto y nuevo usuario:** Hasta que un producto no haya recibido un número considerable de valoraciones el recomendador no es capaz de recomendar dicho producto; por lo tanto, los productos más populares tienden a tomar ventaja de esta situación. Lo mismo sucede con los usuarios recién llegados al sistema. Éstos tendrán que valorar un número considerable de productos antes de que el recomendador pueda realmente entender cuáles son sus preferencias. Por lo tanto, un nuevo usuario que tiene pocas valoraciones, no obtendrá buenas recomendaciones hasta que no haya utilizado el sistema durante un tiempo. Este problema también es conocido como el arranque en frío.
- **Fiabilidad:** Estos sistemas deben ser robustos frente a ataques. Si cualquiera puede recomendar, los propietarios de determinados productos

podrían generar recomendaciones positivas de sus productos y negativas de los de otros. De ahí que se ponga esfuerzo en crear sistemas robustos frente a estos comportamientos no deseados por parte de usuarios.

2.1.2. Recomendadores basados en contenido

Los recomendadores basados en contenido son aquellos que realizan la recomendación en base a la descripción y/o el contenido de los productos que va a recomendar. Este tipo de recomendadores utilizan las características de los productos disponibles junto con las valoraciones asignadas a los productos por un determinado usuario para tratar de recomendar los productos que son similares a los que le gustaron a ese usuario en el pasado. En particular, los diferentes productos candidatos se comparan con los productos que valoró positivamente el usuario y los productos más coincidentes son recomendados. Este enfoque tiene sus raíces en la recuperación de información y en el filtrado de información. En los trabajos presentados por Adomavicius y Tuzhilin (2005), Pazzani y Billsus (2007) y Lops et al. (2011) encontramos una revisión en profundidad de este tipo de recomendadores.

Los recomendadores basados en contenido construyen un perfil de usuario en base a los productos valorados previamente por el propio usuario (Pazzani y Billsus, 2007; Lops et al., 2011). En general, los recomendadores basados en contenido se pueden clasificar de acuerdo a cómo construyen el modelo de usuario a partir de los datos subyacentes, normalmente basándose en técnicas de aprendizaje máquina (Lops et al., 2011; Pazzani y Billsus, 1997; de Gemmis et al., 2008), o utilizando funciones heurísticas normalmente inspiradas en métodos del campo de la recuperación de información (Balabanovic y Shoham, 1997; Diederich y Iofciu, 2006; Cantador et al., 2010).

Uno de los primeros recomendadores basados en contenido fue Fab (Balabanovic y Shoham, 1997) que recomendaba páginas Web a usuarios. El contenido de la página Web estaba representado por las 100 palabras más importantes. De una manera similar, el sistema Syskill & Webert propuesto por Pazzani y Billsus (1997) representaba los productos (que en este caso eran documentos) por las 128 palabras más importantes. La representación de los productos, tal y como lo hacían estos dos primeros sistemas, se conoce con el nombre de modelo del espacio vectorial y es la que más ampliamente ha sido utilizada para representar un producto en este tipo de recomendadores. En este modelo, un producto está representado por el vector de palabras más importantes del mismo. Una de las métricas más extendidas para determinar la “importancia” de una palabra en un documento es la métrica *term frequency/inverse document frequency (TF-IDF)* (Salton et al., 1975).

Bien tengamos los productos representados por las palabras más importantes o por las características del mismo, previamente definidas, el siguiente paso en el proceso de recomendación es recuperar aquellos productos más

similares a los intereses del usuario. Para calcular la similitud entre un producto y el resto de productos se utilizan métricas tales como la similitud del coseno, el método de Rocchio (Rocchio, 1971) o los clasificadores bayesianos que estiman la probabilidad de éxito que tendrá un producto para un determinado usuario (Pazzani, 1999).

Las técnicas que acabamos de mencionar vienen todas del campo de la recuperación de información, pero existen otras técnicas que también son utilizadas con resultados exitosos. Por ejemplo los métodos probabilísticos (Pazzani y Billsus, 1997), las técnicas de clustering, los árboles de decisión y las redes neuronales (Pazzani y Billsus, 2007). Todas estas técnicas coinciden en que no calculan la predicción en base a una métrica heurística, sino que utilizan un modelo de predicción a partir de técnicas estadísticas y de aprendizaje máquina.

Podemos encontrar recomendadores basados en contenido en distintos dominios como puede ser recomendación de noticias, música, películas, etc. En el área de la recomendación de noticias encontramos sistemas como NewsDude (Billsus y Pazzani, 1999), NewT (Sheth y Maes, 1993) o YourNews (Ahn et al., 2007). NewsDude aprende las necesidades a corto plazo del usuario a partir del modelo *TF-IDF* y las preferencias a largo plazo con un clasificador bayesiano a partir de un conjunto inicial de artículos que proporciona el usuario. NewT (Sheth y Maes, 1993) es un recomendador basado en contenido que recomienda artículos de noticias a partir del conjunto de características que los definen. El usuario después de recibir una recomendación puede proporcionar sus preferencias con el artículo propuesto para que así el recomendador pueda ir construyendo el perfil de preferencias del usuario y obtener mejores recomendaciones. De la misma manera, YourNews (Ahn et al., 2007) mantiene perfiles de usuario en base a 8 temas distintos (Nacional, Internacional, Economía, etc.). El perfil de preferencias de los usuarios para cada tema se representa como un vector de términos ponderado extraído del historial de visitas del usuario a las noticias. Se recuperan N artículos del historial de visitas y las 100 palabras con más peso son extraídas para generar el vector final de términos que se utilizará para la recomendación. El proceso de recomendación compara este vector de usuario con los vectores de los nuevos artículos.

El recomendador LIBRA (Mooney y Roy, 2000) implementa un método de categorización bayesiano para la recomendación de libros. Las descripciones de los productos las obtiene de librerías online como Amazon. Citeseer (Bollacker et al., 1998) ayuda al usuario a encontrar artículos científicos. Este recomendador analiza las palabras contenidas en cada artículo junto con las referencias contenidas en el mismo para extraer cuáles pueden ser los artículos similares a uno dado y así generar las recomendaciones. Por último INTIMATE (Mak et al., 2003) es un recomendador de películas que aprende, gracias a las técnicas de categorización de texto, las descripciones

de las películas obtenidas de Internet Movie Database¹. Antes de realizar ninguna recomendación, el usuario debe valorar un número determinado de películas en la siguiente escala: terrible, mala, por debajo de la media, por encima de la media, buena, excelente. Cada vez que el usuario valora una película, ésta es almacenada en su perfil junto con el vector de palabras que la describe. En su trabajo Mak et al. (2003) presentan una comparación de tres algoritmos de aprendizaje máquina para realizar la categorización de los textos: vecinos más cercanos, árboles de decisión y el método Naïve Bayes. Una vez que el clasificador ha sido construido, se utiliza para clasificar las películas no vistas y recomendar aquellas que tengan mejores valoraciones.

Otro trabajo que apuesta por introducir las técnicas de recuperación de información tradicionales en los recomendadores basados en contenido es el trabajo propuesto por Musto (2010). En dicho trabajo presentan un modelo llamado *indexación aleatoria*, que construye el vector de palabras no sólo con los términos que aparecen en los textos sino teniendo en cuenta el contexto en el que se encuentran los términos y estableciendo asociaciones entre términos (palabras que aparecen en un mismo contexto tienden a tener un mismo significado). Para demostrar las bondades de su modelo realizan una evaluación con el dataset de MovieLens y las entradas de las películas contenidas en Wikipedia. *Pharos* es un recomendador basado en contenido para páginas web de tipo social (Zhao et al., 2010). Este sistema tiene una doble funcionalidad, primero detecta el contenido de lo que se *está hablando* en la web y, a continuación, crea un mapa social que informa de los usuarios que hablan de un determinado tema. Además, para un determinado usuario *Pharos* recomienda qué contenido y usuarios le puede interesar seguir. Otro trabajo en la línea del anterior es el presentado en (Hannon et al., 2010), en donde se propone un recomendador con un componente basado en contenido que propone usuarios a los que seguir en Twitter.

La técnica basada en contenido fue la más extendida y utilizada en los primeros recomendadores hasta la explosión definitiva del filtrado colaborativo, pero muestra algunas debilidades como:

- **Sobreespecialización:** La sobreespecialización se da al reducir las recomendaciones a unos contenidos muy similares entre sí sin tener en cuenta la posible arbitrariedad de los gustos e intereses de los usuarios. Aunque los recomendadores que sufren este problema suelen recomendar productos muy relevantes, la utilidad de las recomendaciones se resiente debido a la falta de novedad y variedad en sus productos. El esfuerzo por ganar en relevancia reduce las posibilidades que un usuario tiene de encontrar los productos que en realidad le pueden interesar más si los hubiera conocido. Este problema lo intentan resolver aquellas aproximaciones que introducen diversidad, novedad y sorpresa (*serendipity*) en la recomendación.

¹<http://www.imdb.com>

- **Restricciones sobre el contenido:** Los recomendadores basados en contenido dependen de las características disponibles asociadas explícitamente a los productos. Estas características deben poder ser interpretadas automáticamente por algún sistema o deben ser extraídas manualmente dificultando así el proceso de mantenimiento si el dominio es complejo o el volumen de productos es grande.
- **Nuevo usuario:** De nuevo nos encontramos con este problema, un usuario recién llegado al sistema tiene que efectuar un número mínimo de valoraciones antes de que el recomendador pueda construir un modelo de usuario para él que sea fiable.

2.1.3. Recomendadores basados en conocimiento

Aunque los recomendadores más utilizados y más conocidos son los colaborativos y los basados en contenido, no en todas las situaciones son los más adecuados. Por ejemplo, los recomendadores colaborativos necesitan partir de una base de datos de valoraciones de los usuarios sobre los productos disponibles para poder realizar recomendaciones precisas y acertadas a cualquiera de estos usuarios. Los basados en contenido buscan nuevos productos a recomendar basándose en las similitudes con los valorados por el usuario en el pasado. Por tanto, estos sistemas requieren que el usuario haya valorado un número mínimo de productos para poder construir su perfil de usuario y así realizar las recomendaciones adecuadas a su proceso de búsqueda de nuevos productos. En situaciones reales nos encontramos ocasiones en las que los modelos anteriores no son aplicables, bien porque no disponemos de las valoraciones suficientes o bien porque no tenemos un conocimiento profundo de los usuarios. Para solventar este tipo de situaciones tenemos el último tipo de recomendadores, los recomendadores basados en conocimiento. Este tipo de recomendadores utiliza toda la información disponible sobre las características del usuario, de los productos y el conocimiento del dominio para efectuar las recomendaciones (Burke, 2000). Este conocimiento permite al sistema de recomendación razonar con criterios semánticos e interpretar los nuevos episodios de recomendación de forma flexible y contextualizada. La inclusión de conocimiento adicional en los sistemas de información, en general, y en los sistemas de recomendación, en particular, ayuda en las tareas de aprendizaje de perfiles de usuario para la personalización de los resultados, así como en tareas de recuperación aproximada que requiera valoraciones precisas de los niveles de similitud y de diferencia existentes entre dos entidades. En el caso particular de los sistemas de recomendación, este conocimiento alivia también problemas tradicionales de estos sistemas, como el arranque en frío o la dispersión de datos, ya que este conocimiento adicional permite el razonamiento y la inferencia de información a partir de la existente.

Según la clasificación propuesta por Jannach et al. (2010) dentro de los recomendadores basados en conocimiento tenemos dos tipos: recomendadores basados en casos (Smyth, 2007) y recomendadores basados en restricciones (Felfernig y Burke, 2008; Zanker et al., 2010). Ambas técnicas son similares en cuanto al proceso de recomendación: el usuario debe especificar sus requisitos y el sistema trata de conseguir una solución. Si no es posible encontrar una solución el usuario deberá modificar sus requisitos. Las diferencias entre ambas técnicas radica en la manera en la que cada una utiliza el conocimiento. Las aproximaciones basadas en casos se centran en recuperar productos similares a partir de distintas métricas de similitud. Por su parte las aproximaciones basadas en restricciones hacen uso de un conjunto de reglas de recomendación previamente definidas y el conjunto de productos recuperados se determina a partir de aquellos productos que cumplen las reglas de la recomendación.

Las primeras aproximaciones de recomendadores basados en casos utilizaban una aproximación basada en consulta en la que el usuario debía de especificar sus requisitos (Burke, 2002b). Uno de los primeros ejemplos de sistema recomendador basado en casos fue el empleado por Analog Devices (Vollrath et al., 1998), uno de los principales fabricantes y distribuidores de componentes electrónicos en EEUU. Este sistema está pensado para recomendar dispositivos electrónicos (amplificadores operacionales principalmente). El usuario debía rellenar un extenso formulario indicando los parámetros del dispositivo que estaba buscando. Una vez terminado, el sistema presentaba al usuario un conjunto de dispositivos que más se asemejan a los requisitos especificados. Cada caso se corresponde con un dispositivo y está formado por cerca de 40 parámetros, tales como especificaciones de señal de entrada, señal de salida, dimensiones, funcionalidad y muchos más. Sermo (Bridge, 2002), Personal Travel Assistant (Coyle et al., 2002) y CASPER (Bradley et al., 2000) siguen también una aproximación basada en casos. El primero ayuda al usuario a elegir una casa de alquiler vacacional, el segundo recomienda viajes, y, por último, CASPER es un sistema de contratación en línea que sugiere posibles trabajos a un usuario.

La formulación de consultas puede resultar complicada si el usuario no tiene un conocimiento suficiente del dominio o no sabe bien lo que quiere. Este inconveniente dio lugar a plantear un nuevo esquema de recomendación que ayudara a los usuarios a navegar por el espacio de productos con el fin de encontrar aquellos productos que mejor se adaptaran a sus necesidades. Con este objetivo nacieron las técnicas de conversación basadas en críticas.

Las críticas son una manera efectiva para recoger los requisitos del usuario ya que éste especifica sus intereses en forma de objetivos no conseguidos (Burke, 2000). Si por ejemplo estamos buscando un restaurante, una crítica sobre el precio del restaurante mostrado –un precio más bajo– puede ser activada y así reactivar la búsqueda de un mejor restaurante. Un ejemplo

muy claro de este tipo de recomendadores es *Entree* (Burke et al., 1996), un recomendador que sugiere restaurantes en la ciudad de Chicago. *Entree* pertenece a la familia de los sistemas *FindMe* (Burke et al., 1997), que fueron de los primeros recomendadores que consiguieron incluir realimentación por parte del usuario en forma de críticas a las recomendaciones propuestas. Otros de los sistemas que pertenecen a esta familia pero con dominios de aplicación distintos son: Car Navigator (Burke et al., 1997), que ayuda al usuario a elegir un coche nuevo; Video Navigator y PickAFlick (Burke et al., 1997), que recomiendan películas para alquilar; RentMe (Burke et al., 1997), que ayuda a encontrar apartamentos; y por último Kenwood (Burke et al., 1997), que recomienda configuraciones para un sistema de sonido en casa. El proceso de recuperación en todos los sistemas *FindMe* se realiza de una manera similar. Tienen una base de productos, recuperan de ella aquellos que cumplen las restricciones impuestas por la consulta y son ordenados en función del criterio establecido en cada sistema concreto. En el caso de *Entree*, el usuario debe rellenar un formulario indicando características concretas de cómo le gustaría que fuera el restaurante (tipo de cocina, precio, etc..) o puede indicar el nombre del restaurante que busca. Junto con la primera recomendación aparecen una serie de críticas que puede realizar a cada restaurante tales como un restaurante mejor, o menos caro, uno que sea más tradicional o más creativo, más tranquilo o más animado, y además puede decir un restaurante similar al mostrado pero con un tipo de cocina diferente. Después de que el usuario realice su crítica, ésta actuará como filtro para la siguiente recuperación, obteniendo así un subconjunto de productos a partir del conjunto anterior con los restaurantes que cumplen las nuevas restricciones. En algunos casos, el resultado de aplicar este filtro ocasiona que el conjunto sea vacío. En esta situación *Entree* le dice al usuario que no existen restaurantes con las características pedidas. Existen otros trabajos como los presentados en (Salamó et al., 2005; Pu et al., 2008) que muestran las ventajas de utilizar sistemas de recomendación basados en conocimiento con críticas en el campo de la recomendación de viajes y el comercio electrónico.

Adaptive Place Advisor (Thompson et al., 2004) es un recomendador de restaurantes en la zona de la bahía de San Francisco, EEUU. En este caso, los productos a recomendar son los diferentes restaurantes descritos por atributos tales como el tipo de cocina, la valoración, el precio, la situación, si admite reservas, si tiene parking y las opciones de pago. Su principal característica es que utiliza otro tipo de navegación conversacional: la navegación por preguntas. Estas preguntas las puede realizar tanto el sistema como el usuario, de ahí que el módulo de procesamiento de lenguaje natural que maneja este sistema sea una parte fundamental. Para enriquecer el modelo de usuario, éste guarda la información acerca de las preguntas que el usuario prefiere responder, así como la tendencia de respuesta a cada una de ellas. Esta preferencia de preguntas se almacena en forma de un valor probabilísti-

co en cada atributo entre 0 y 1. Es decir, si un usuario habitualmente prefiere preguntar por el tipo de cocina, el sistema asignará un valor probabilístico mayor a este atributo, para que en futuras conversaciones sea de los más significativos.

La recomendación basada en restricciones da soporte a los clientes explicando los elementos, y proponiendo mejoras en aquellas situaciones en las que no ha sido posible encontrar una solución (satisfacer la restricción). De ahí que estos recomendadores sean ampliamente utilizados en dominios complejos, como pueden ser los servicios financieros. Un ejemplo de esto es el sistema VITA (Felfernig et al., 2007), un recomendador de servicios financieros que utiliza un perfil de usuario que indica cuáles son sus requisitos, las características de los productos financieros y un conjunto de restricciones que indican qué productos pueden ser recomendados según el contexto, para generar las recomendaciones. En (Ricci et al., 2002) encontramos el recomendador Intelligent Travel Recommender, donde se explora un modelo de recomendación basado en críticas que permite la selección de lugares de vacaciones, actividades y atracciones, y permite crear un plan de viaje personalizado. Un plan de viaje personalizado se realiza combinando la iniciativa del usuario y del sistema: el usuario plantea consultas y el recomendador le propone recomendaciones. Este sistema, en caso de no obtener un conjunto que satisfaga la consulta del usuario, ayuda al usuario explicándole qué otros resultados podría obtener realizando algunas relajaciones en la consulta.

Por último, presentamos Advisor Suite, un recomendador para venta *online* (Jannach, 2004) que evalúa las restricciones en forma de expresiones *si...entonces...* y las aplica como filtro. En este sistema las dependencias entre los requisitos de usuario y las características de los productos están explícitamente representadas. Las preferencias del usuario se recogen a través de una navegación por preguntas.

Al igual que las técnicas anteriores, la recomendación basada en conocimiento también muestra debilidades:

- La principal de todas es la dificultad que existe para adquirir el conocimiento del dominio. Para conseguir unas recomendaciones de calidad debe hacerse un esfuerzo previo en conseguir un conocimiento lo más completo posible.
- La sobreespecialización en las recomendaciones también está presente en este tipo de sistemas. Las recomendaciones proporcionadas por los recomendadores que utilizan una aproximación tanto basada en casos como basada en restricciones tienen el peligro de ser muy similares entre sí.
- El problema del conjunto vacío para los recomendadores basados en conocimiento que utilizan un modelo basado en restricciones. En este

caso el sistema debería ayudar al usuario a reformular su consulta o proponer una recomendación que cumple una parte de las restricciones.

2.1.4. Estrategias de recomendación híbrida

Cada una de las técnicas de recomendación vistas produce buenos resultados aunque también tienen sus puntos débiles, por ejemplo el problema del arranque en frío en los recomendadores colaborativos o el problema de la adquisición de conocimiento para los recomendadores basados en conocimiento. Con el objetivo de aprovechar mejor las fortalezas de cada técnica de recomendación y minimizar las debilidades surgieron las estrategias de recomendación híbrida. Estas estrategias combinan sistemas de recomendación desarrollados siguiendo las técnicas anteriormente descritas para obtener así mejores resultados.

En estos enfoques híbridos, la decisión más importante es cómo combinar los distintos recomendadores a utilizar. En primer lugar, entonces, hay que decidir qué tipo de recomendadores se van a combinar y qué información va a utilizar cada uno. Un enfoque estándar descrito en la literatura consiste en combinar recomendadores basados en contenido con recomendadores basados en filtrado colaborativo para eliminar la dispersión y la falta de información sobre las características. Sin embargo, otros tipos de recomendadores y fuentes de información, como contactos y marcas temporales, han sido recientemente integradas en las formulaciones clásicas de las técnicas estándar.

Una primera aproximación a los tipos y características de los recomendadores híbridos fue la realizada por Burke (2002a). En ella podemos encontrar hasta siete métodos diferentes para combinar estrategias de recomendación, a saber:

- **Ponderado.** Las valoraciones proporcionadas por los recomendadores se agregan utilizando una combinación lineal o un esquema de votación para obtener así la puntuación final para cada producto. El producto (o subconjunto de productos) con mayor puntuación será el que se ofrezca al usuario. Cada uno de los recomendadores pueden tener asociados distintos pesos en la combinación final. Un ejemplo de recomendador híbrido ponderado es *Movie Recommender*, descrito en Mobasher et al. (2003). Se trata de un recomendador de películas con dos componentes: uno que utiliza técnicas de filtrado colaborativo para así comparar entre perfiles de usuario la similitud entre sus preferencias; el otro usa la información sobre las características de las películas y recomienda aquellas cuyas características coinciden con los gustos del usuario. La salida de estos dos componentes se combina usando una función ponderada lineal.
- **Cascada.** Un sistema híbrido en cascada funciona como un proceso

secuencial. Primero se ejecuta una de las estrategias de recomendación y obtiene un primer conjunto de productos candidatos a ser incluidos en la recomendación final. A continuación, el primer conjunto de candidatos sigue el proceso y las siguientes estrategias refinan la recomendación seleccionando sólo algunas de las sugerencias obtenidas en la etapa anterior y así hasta llegar a la última estrategia. Como ejemplo de recomendador en cascada tenemos *EntreeC* (Burke, 2007), creado añadiendo a *Entree* (Burke et al., 1996) un módulo colaborativo de marcaje-ordenación para aquellos productos con iguales valores. Este recomendador soluciona un problema encontrado en su predecesor: la devolución de muchos productos con iguales valoraciones, los cuales no podían ser ordenados en relación con el resto de productos.

- **Conmutación.** En este caso, en lugar de ejecutar todas las estrategias simultáneamente, el sistema emplea algún criterio para conmutar entre ellas. Cuando se cumplen ciertas condiciones el sistema emplea una estrategia y, en caso contrario, recurre a la(s) restante(s). Un ejemplo de sistema híbrido que utiliza conmutación es *NewsDude*, un recomendador que ofrece artículos de noticias a los usuarios (Billsus y Pazzani, 1999). Este sistema se compone de tres recomendadores: dos basados en contenido y uno colaborativo. Los tres componentes están ordenados. Si el primero no produce una recomendación fiable, se pasa al segundo, y así hasta el tercero. Zanker y Jessenitschnig (2009a) proponen un recomendador conmutado que en realidad elige entre dos variantes híbridas que a su vez combinan una estrategia de filtrado colaborativo con otra basada en conocimiento. El algoritmo conmutado actúa de manera que si la primera variante híbrida (que opera en cascada) devuelve menos de un cierto número de recomendaciones, entonces elige el segundo sistema híbrido (que opera con una aproximación ponderada) como plan alternativo.
- **Mixto.** Este esquema reúne en una misma recomendación productos que han sido sugeridos mediante las diferentes estrategias que componen el sistema híbrido. Un ejemplo de sistema híbrido mixto es *PTV-Plus*, un recomendador de programas en televisión digital (Smyth y Cotter, 2001). *PTVPlus* se compone de dos recomendadores, uno basado en contenido y otro colaborativo, en el que el resultado final será una combinación de los resultados producidos por ambos.
- **Combinación de características.** En este modelo se funden en un único conjunto de datos los distintos datos que utilizan las diferentes estrategias, y con este nuevo conjunto se ejecuta un único algoritmo de recomendación. Por ejemplo, Basu et al. (1998) proponen un recomendador de películas basado en contenido que construye los modelos de usuario y a continuación las valoraciones de los usuarios reflejadas en

el modelo construido son combinadas con las características de los productos, obteniendo así un único conjunto de datos que será utilizado en el algoritmo de recomendación. Otra aproximación que utiliza combinación de características es la propuesta por Zanker y Jessenitschnig (2009b). En ella utilizan diferentes maneras de obtener las valoraciones en función de la predicción de la precisión y de la disponibilidad para posteriormente utilizar todos estos datos en un único algoritmo de recomendación. El trabajo propuesto por Tsatsou et al. (2009) presenta un recomendador híbrido que combina el contenido de los productos a recomendar –basado en contenido– con el conocimiento del dominio reflejado en una ontología –basado en conocimiento. El primero extrae de los productos que ha visitado el usuario la información textual. De este contenido se extrae la información semántica y se clasifica en una ontología –que previamente ha sido definida y poblada. El sistema realizará un modelo de usuario gracias a la ontología y así podrá recomendar aquellos productos –en este caso anuncios– que mejor se adapten a sus preferencias. Por su parte Gunawardana y Meek (2009) proponen utilizar un recomendador híbrido a partir de uno basado en contenido y uno colaborativo utilizando una máquina de Boltzman para unificar los resultados de ambos recomendadores. El modelo propuesto codifica la información de ambos recomendadores como si fueran características y después aprende los pesos que debe tener cada característica para obtener la mejor recomendación. Con este método la ponderación se realiza de manera automática. En su trabajo presentan una evaluación en los dominios de la música y la venta *online* obteniendo muy buenos resultados de eficiencia y precisión y evitando el problema del arranque en frío de los recomendadores colaborativos.

- **Incorporación de características.** En este esquema, una de las estrategias genera una recomendación junto con una característica nueva para el producto, por ejemplo, una característica del tipo: “productos relacionados”. A continuación, esa información se incorpora como una característica más de ese producto para las otras técnicas de recomendación. Melville et al. (2002) describen un recomendador híbrido que, primero, a través de un recomendador basado en contenido y un conjunto de datos de entrenamiento, genera valoraciones para usuarios que todavía no disponen de éstas en los productos del conjunto de datos de entrenamiento. Tanto las valoraciones generadas como las asignadas directamente por los usuarios son utilizadas por un recomendador colaborativo. En (Mooney y Roy, 2000) se describe un sistema de recomendación de libros basado en contenido. Este sistema de recomendación extrae información sobre libros a partir de la información que aparece en Amazon. Esta información contiene las recomendaciones que Amazon hace al usuario, del tipo “Títulos relacionados” o “Autores

que también pueden ser de interés”. Así pues, el sistema de recomendación con incorporación de características, además de obtener como productos de entrada para el aprendizaje del sistema las propiedades del producto recomendado (como puede ser autor de la obra, tipo de obra o número de páginas), también incorpora las recomendaciones, en este caso del tipo “Títulos relacionados” o “Autores que también pueden ser de interés”, a su motor de aprendizaje.

- **Metanivel.** En este caso, el modelo completo generado por una de las estrategias se utiliza como entrada en las otras existentes en el recomendador híbrido. Como aclara Burke (2002a): *“La diferencia entre este método de recomendación híbrida y el basado en incorporación de características es que, en este último, el modelo aprendido sólo se utiliza para generar valores de características que se usan como entrada en las siguientes estrategias, mientras que en el metanivel se utiliza todo el modelo como dato de entrada”*. Fab (Balabanovic, 1997) es un recomendador de documentos que usa la estructura “collaboration through content” propuesta por Pazzani (1999), que utiliza un recomendador basado en contenido para construir los modelos de usuario y después un módulo colaborativo que filtra los usuarios.

Recientemente Jannach et al. (2010) han descrito una clasificación similar de los recomendadores híbridos. En esta clasificación se entiende que las siete categorías descritas por Burke (2002a) en realidad puede verse como tres categorías que engloban a todas las demás: monolíticos, paralelos y segmentados.

Como hemos podido ver existen diversas alternativas para diseñar un recomendador híbrido. No podemos decir cuál es mejor que otra, ya que las características del dominio y el proceso de recomendación empleado nos indicarán cuál puede ser la mejor estrategia de combinación. Lo que sí parece claro es que las estrategias de recomendación básicas pueden mejorarse gracias a los diseños híbridos. Un buen ejemplo de esta afirmación es la competición Netflix Prize.² Esta competición fue llevada a cabo entre los años 2006 y 2011 y condujo a cientos de estudiantes e investigadores a mejorar un algoritmo de filtrado colaborativo para películas. Todos los primeros clasificados utilizaron un amplio número de recomendadores ensamblados entre sí que pueden ser considerados como técnicas híbridas. El objetivo era mejorar la precisión global del sistema de recomendación. Fruto de esta competición fue la proliferación de los sistemas de recomendación híbridos.

²Visitar <http://www.netflixprize.com> para más información

2.2. Líneas de investigación abiertas en los sistemas de recomendación

El estudio de los sistemas de recomendación se centró durante mucho tiempo en mejorar la calidad de las recomendaciones a través de la precisión de los resultados obtenidos. Sin embargo, algunos trabajos recientes han hecho notar que no sólo la precisión es un aspecto importante en la calidad de una recomendación sino que también lo es la experiencia del usuario con los resultados obtenidos (McNee et al., 2006). Con el objetivo de mejorar las recomendaciones, otras características como la diversidad de los resultados en la recomendación, la confianza en las recomendaciones proporcionadas, la explotación de la información contextual y el esfuerzo del usuario en conseguir un resultado deseable, se han convertido en aspectos importantes a tener en cuenta en el diseño de un sistema de recomendación.

En esta sección nos centraremos en presentar tres de estos aspectos que se han hecho un hueco en la investigación en los sistemas de recomendación (Ricci et al., 2011): la diversidad, la información contextual y alternativas de interacción usuario-recomendador. En un primer momento hablaremos de algunos trabajos que pretenden mejorar la calidad de las recomendaciones introduciendo diversidad en el conjunto de productos recomendados. Nos parece una característica interesante a tener en cuenta de cara al trabajo de esta tesis ya que la diversidad busca mejorar la calidad global de las recomendaciones desde el punto de vista del usuario evitando la monotonía y promoviendo que los usuarios reciban sugerencias que acierten en sus gustos, pero que ofrezcan además un amplio abanico de alternativas. La segunda característica que presentaremos es la explotación de la información contextual. Según el diccionario de la lengua española, se entiende por contexto el entorno físico –como el tiempo, localización– o de situación en el cual se considera un hecho. Incluir información adicional sobre el contexto en los sistemas de recomendación puede ser un gran valor para las recomendaciones ya que añade un nivel más de personalización, considerando la influencia del ambiente del usuario en su apreciación de los productos que se recomiendan (Adomavicius et al., 2005). Finalmente, presentaremos modelos de interacción con el usuario que buscan aliviar la carga de trabajo que puede suponer para un usuario realizar consultas sobre los productos que le gustaría que el sistema le recomendara.

2.2.1. Incorporación de diversidad en el proceso de recomendación

La tarea de recomendación ha estado centrada en proporcionar productos cuya probabilidad individual de resultar relevantes para un usuario es alta. Este objetivo por sí solo no resulta necesariamente útil para el propósito

de la recomendación para el usuario final. Por ejemplo, imaginemos que un usuario desea encontrar un destino vacacional de playa para el mes de mayo en España, que tenga buen ambiente por la noche y lugares de recreo. Con bastante probabilidad las primeras recomendaciones podrían estar situadas en la Costa del Sol. Si el usuario descarta este destino, es probable que dentro de las opciones mostradas no se encuentren alternativas: todos los productos recomendados son muy similares a la consulta y también lo serán entre ellos.

La diversidad busca mejorar la calidad global de las recomendaciones desde el punto de vista del usuario evitando la monotonía y promoviendo que los usuarios reciban sugerencias que acierten en sus gustos, pero que ofrezcan además distintas alternativas de elección. Esta perspectiva contrasta con el planteamiento tradicional, que plantea como único objetivo la acumulación de relevancia individual de los elementos a recomendar. La diversidad, por tanto, es una característica que nos habla de la variedad que encontraremos entre los productos recomendados.

Una buena recomendación, aparte de proporcionar variedad entre los productos, debería tener en cuenta de manera explícita los múltiples intereses del usuario reflejados en el perfil del mismo de manera que la estrategia de recomendación pueda proporcionar al usuario no sólo recomendaciones diversas sino mejor ajustadas al usuario, sus características, y la variedad de sus gustos.

Los primeros métodos para introducir diversidad en el proceso de recomendación proponen tener en cuenta alguna medida de la misma al tiempo que se tiene en cuenta la similitud con los intereses del usuario. De esta forma se prima no sólo a los productos que, por similitud, permiten satisfacer mejor los intereses o preferencias del usuario sino aquellos que, al mismo tiempo, permiten que el conjunto de productos sugeridos finalmente sea lo más diverso posible.

El primer trabajo que introdujo explícitamente la diversidad fue el propuesto por Smyth y McClave (2001). En dicho trabajo se proponen tres alternativas para recomendar k productos a partir de un conjunto C de productos existentes y una consulta formulada por el usuario. Las tres alternativas (*Bounded Random Selection*, *Greedy Selection* y *Bounded Greedy Selection*) mantienen el compromiso con la similitud con la consulta, pero a la vez introducen diversidad.

La estrategia más simple que intenta introducir diversidad al conjunto recuperado es la *selección aleatoria* (*Bounded Random Selection*), que consiste en seleccionar k productos aleatoriamente a partir de un conjunto de $b \cdot k$ productos similares a la consulta, con $b > 1$.

Una mejor opción para incluir diversidad y mantener el compromiso con la similitud es el método de *selección voraz* (*Greedy Selection*). Este método construye incrementalmente el conjunto R que al final contendrá los k productos a recomendar. En cada paso, los restantes productos existentes

en C que aún no han sido considerados para ser recomendados son ordenados en función de su calidad, y aquel con mayor calidad se añadirá al conjunto de productos seleccionados para ser recomendados hasta ese momento $R = \{r_1, r_2, \dots, r_m\}$. La clave de este método es incluir una métrica de calidad que combina similitud y diversidad. Así, la calidad de un producto c candidato a ser incluido en R es proporcional a la similitud entre c y la consulta Q ($Similitud(Q, c)$), y la diversidad de c relativa a los productos que ya se encuentran en R ($DiversidadRel(c, R)$) :

$$Calidad(Q, c, R) = Similitud(Q, c) * DiversidadRel(c, R) \quad (2.1)$$

donde

$$DiversidadRel(c, R) = \begin{cases} 1 & \text{si } R = \{\}; \\ \frac{\sum_{i=1..m} (1 - Similitud(c, r_i))}{m} & \text{en otro caso} \end{cases} \quad (2.2)$$

En el algoritmo descrito, se elige como primer producto a incluir en R el más similar a la consulta. El problema de este algoritmo es que tiene un coste computacional muy elevado.

Para reducir la complejidad de cálculos impuesta por la anterior alternativa está la *Selección voraz acotada* (*Bounded Greedy Selection*). En este caso el algoritmo primero selecciona los $b \cdot k$ productos más similares a la consulta ($b > 1$) y luego aplica sobre éstos la selección voraz antes explicada. Por supuesto esta ganancia de eficiencia tiene un coste: podemos perder algún producto que aún teniendo una menor similitud con la consulta tiene una gran diversidad con el resto de productos, lo cual compensaría la baja similitud y haría que pudiese entrar en el conjunto de productos elegidos para ser mostrados al usuario.

Otro enfoque interesante es el algoritmo DCR-1 propuesto en (McSherry, 2002). En este trabajo, ante una consulta Q formulada por el usuario la colección de productos se divide en capas de similitud. En cada capa se agrupan aquellos productos que tienen igual similitud con respecto a Q . La primera capa es la de mayor similitud, esto es, aquella cuyos productos son los que mejor satisfacen Q . Cada capa contendrá los productos que satisfacen Q mejor que lo que lo hacen los productos de la capa siguiente.

Una vez que han sido reorganizados todos los productos en capas, el algoritmo DCR-1 se encarga de seleccionar los k productos que serán mostrados al usuario fomentando la existencia de diversidad. Este algoritmo actúa en tres pasos:

1. Paso 1: determina el conjunto de productos candidatos por similitud (*CPS*) que serían recomendados utilizando una métrica de similitud pura.

2. Paso 2: identifica la capa de menor similitud que contribuye a la formación de *CPS* (es decir, la capa en la que se encuentra el producto menos similar de los que constituyen *CPS*). Usando un ejemplo tremendamente simplificado, si hubiésemos obtenido tres capas de similitud L_1 , L_2 y L_3 con dos productos en L_1 , tres productos en L_2 y uno en L_3 , y asumiendo que $k = 4$, la capa de menor similitud que contribuiría a la formación de *CPS* sería L_2 .
3. Paso 3: ejecuta un algoritmo auxiliar que, dado un conjunto inicial de productos, un conjunto de candidatos a añadir a ese conjunto inicial y un valor de k , consigue un conjunto *ConjRecomendados* de productos que será el que definitivamente se presente al usuario. *ConjRecomendados* contendrá todos los productos del conjunto inicial más algunos del conjunto de candidatos a ser recomendados (que se van añadiendo de forma iterativa). La decisión de qué producto del conjunto de candidatos se añade en cada paso del algoritmo se basa en localizar el producto del conjunto de candidatos que tiene mayor diversidad relativa con los que ya se encuentran en *ConjRecomendados* en ese momento. El conjunto inicial estará formado por todos los productos que se encuentran en capas de similitud mayor que la capa identificada como la de menor similitud que contribuye a la formación de *CPS*, siempre que el producto de mayor similitud con la consulta, P_{max} , no se encuentre en la capa identificada como de menor similitud que contribuye a la formación de *CPS*. Si, por el contrario, P_{max} se encuentra en la capa identificada como la de menor similitud que contribuye a la formación de *CPS*, entonces el conjunto inicial es P_{max} . El conjunto de candidatos a añadir al conjunto inicial estará formado por todos los productos de la capa identificada como la de menor similitud que contribuye a la formación de *CPS* si P_{max} no se encuentra en la capa de menor similitud. Si por el contrario P_{max} sí se encuentra en la capa de menor similitud que contribuye a la formación de *CPS*, el conjunto de candidatos serán todos los demás productos de dicha capa. En el ejemplo simplificado usado en el paso anterior, donde $k = 4$, el conjunto inicial estaría formado por los dos productos de la capa L_1 . Por su parte, el conjunto de candidatos a añadir estaría formado por los tres productos en L_2 , de los cuales se elegirían dos que sean los más diversos en relación con los de la capa L_1 .

Esta aproximación también permite mejorar la diversidad preservando la similitud con la consulta del usuario. Sin embargo, la mejora en la diversidad es menor que con el algoritmo de selección voraz acotada explicado anteriormente, ya que todos los productos de las capas de mayor similitud son siempre incluidos en el conjunto resultado sin ninguna mejora en la diversidad. La segunda alternativa que proponen en (McSherry, 2002) está basada en la noción de “intervalos de similitud”, que abarcan cada uno más

de una capa de similitud de las usadas por DCR-1. La ventaja de esta aproximación es que se puede mejorar la diversidad de los resultados relajando las restricciones de similitud con la consulta. La similitud con la consulta se reduce pero a un nivel tolerable definido por el ancho de los intervalos de similitud.

ExpertClerk (Shimazu, 2002) también apuesta por incluir diversidad en sus recomendaciones. En este trabajo el sistema propone 3 productos seleccionados a partir del conjunto $R = \{r_1, \dots, r_k\}$ de productos recuperados para una consulta del usuario. Primeramente se selecciona para la lista final de recomendación aquel producto de R que más cerca se encuentra de la mediana de dicho conjunto, donde la mediana se calcula con la siguiente fórmula:

$$r_{med} = \left(\frac{1}{k} \sum_{j=1}^k f_{j1}, \frac{1}{k} \sum_{j=1}^k f_{j2}, \dots, \frac{1}{k} \sum_{j=1}^k f_{jn} \right) \quad (2.3)$$

donde $r_j = \{f_{j1}, f_{j2}, \dots, f_{jn}\}$ es el conjunto de los valores de los atributos del producto r_j . A continuación se generan las características positivas y negativas de cada producto que aún hay en R y en función de ellas se seleccionan el segundo y tercer producto que serán recomendados. Para cada atributo de cada producto de R se calcula la distancia entre el valor de dicho atributo en ese producto y el valor de dicho atributo en la mediana. Si la distancia computada para un atributo supera un umbral predefinido, se considera que ese atributo es una característica del producto. Si el valor de esa característica es mayor que el valor del mismo atributo en la mediana, se considera una característica positiva y si es menor se considera una característica negativa. El número total de características de un producto será la suma de todas las características, positivas y negativas. El producto que tiene el mayor número de características será seleccionado como segundo producto, y el segundo producto con mayor número de características será presentado como tercera opción.

Cabe destacar el trabajo de Ziegler et al. (2005) en el que, además de introducir diversidad, realizan una evaluación con usuarios sobre la percepción de la diversidad y cómo ésta afecta a la satisfacción con las recomendaciones obtenidas. Los autores definen la diversidad como la disimilitud promedio entre los productos de una lista de recomendación, y proponen como solución un algoritmo de reordenación voraz que busca un compromiso entre máximo acierto individual y mínima distancia con productos previamente ordenados. La manera que Ziegler et al. (2005) emplean para promover la diversidad en la lista de productos recomendados consiste en diversificar los temas contenidos en una taxonomía. Los autores proponen una métrica de similitud basada en la taxonomía y la utilizan para calcular la similitud dentro de la lista (*intra-list similarity*, ILS).

El algoritmo *topic-diversification* reordena una lista de recomendación.

Dicho algoritmo necesita definir una métrica de similitud entre productos que devuelva un valor en el intervalo $[-1, +1]$ que puede ser extraído por medio de, por ejemplo, una taxonomía de las características de los productos. Básicamente, el algoritmo consiste en la elección voraz del producto que minimice

$$(1 - \lambda) \text{rank}_R(i) + \lambda \text{rank}_{\text{sim}^*(S, \cdot)}(i) \quad (2.4)$$

donde λ es el factor de diversificación, rank_R devuelve la posición en la lista original del producto i y $\text{rank}_{\text{sim}^*(S, \cdot)}$ devuelve la posición en la clasificación resultante de ordenar los elementos de $R \setminus S$ (donde S es la lista ordenada) por su similitud con los elementos en S en orden descendente. Los autores sugieren que su algoritmo se asemeja a la permeabilidad selectiva de la membrana de la biología molecular. Este enfoque también es muy similar al esquema “Maximal Marginal Relevance” propuesto en Recuperación de Información (IR) para la diversificación de búsqueda y resumen automático (Carbonell y Goldstein, 1998).

La evaluación del algoritmo de recomendación con diversidad propuesto por (Ziegler et al., 2005) la realizan en un recomendador con filtrado colaborativo de libros. En esta evaluación comparan no sólo la precisión en diferentes niveles de la diversificación, tanto para filtrado colaborativo basado en usuario como basado en producto, sino también los resultados de satisfacción subjetiva obtenidos a partir de una encuesta realizada a usuarios. Sus resultados muestran que la satisfacción de los usuarios con una lista de recomendaciones no sólo depende de la precisión de la misma, sino también de la gama de temas de lectura cubiertos. Las encuestas muestran que aunque las listas con mayores niveles de diversidad eran menos precisas, eran más preferidas por los usuarios. También muestran que la satisfacción del usuario con las recomendaciones era mayor si éste era capaz de percibir la diversidad obtenida.

Otro trabajo en la línea del anterior, en el que realizan una evaluación de la percepción de la diversidad con usuarios, es el propuesto por Castagnos et al. (2010). En este trabajo los autores investigan el impacto de los recomendadores en los patrones de búsqueda de los usuarios. Esto lo realizan mediante la observación de los comportamientos de interacción en una web de venta con un sistema de seguimiento de la vista (*eye-tracking*). Esta observación fue realizada en una sesión experimental durante tres semanas con un grupo controlado de usuarios. Estos usuarios utilizaron un sistema de compra online de perfumes que incorporaba el recomendador y posteriormente contestaron un cuestionario de 14 preguntas. Junto con las respuestas proporcionadas por los usuarios analizaron los registros de actividad almacenados por el sistema (categorías de productos visitadas, procedencia del producto añadido a la cesta de la compra, páginas consultadas, etc.). En su investigación señalan que las recomendaciones con diversidad aumentan la confianza de los usuarios en el sistema de compra debido a que si los

productos recomendados son diversos entre sí, los usuarios pueden realizar comparaciones entre los productos y así elegir el mejor para ellos.

Zhang y Hurley (2008) presentan un trabajo en el que pretenden introducir diversidad en recomendadores colaborativos basados en usuario. La premisa de estos sistemas, como vimos en la Sección 2.1.1.1, es que usuarios similares tienen intereses similares, por lo tanto la similitud debe ser la principal característica que guíe el proceso de recomendación –obtener los productos que usuarios similares a nosotros valoraron. Zhang y Hurley (2008) plantean el problema de la diversidad como un problema de optimización. Dada la matriz de distancias $n \times n$ de los N productos con mejor predicción para el usuario actual, el objetivo es encontrar un vector binario de tamaño n que describa cuáles son los productos que pertenecen al subconjunto más diverso de tamaño k ($k < N$).

También existen algunos trabajos que introducen diversidad en recomendadores conversacionales basados en críticas. Los trabajos de McCarthy et al. (2005) y (Chen y Pu, 2007) utilizan la diversidad para proponer las críticas que un usuario puede hacer sobre los productos recomendados. El objetivo de introducir diversidad en las críticas es guiar al usuario hacia otros tipos de productos que, si no fuera por la diversidad, nunca llegarían a ser recomendados. En ambos trabajos proponen sendas métricas de calidad para las críticas en las que se premian aquellas críticas que son más diversas con respecto a las ya propuestas.

2.2.2. Estrategias de recomendación sensibles al contexto

La definición de contexto en los sistemas de recomendación más ampliamente aceptada es la propuesta por Jannach et al. (2010) en la que entienden por contexto los parámetros de la situación que pueden ser conocidos por el sistema y que pueden tener un impacto en la selección y clasificación de los resultados de recomendación. Schilit et al. (1994) enumeran los principales aspectos del contexto como dónde estás, con quién estás y qué recursos están cerca de ti.

El uso de la información contextual en los sistemas de recomendación fue introducido ligeramente en (Herlocker y Konstan, 2001), donde los autores plantean como hipótesis que la inclusión de los conocimientos sobre la tarea que va a realizar el usuario en el algoritmo de recomendación en ciertas aplicaciones puede llevar a mejores recomendaciones. Sin embargo no ha sido hasta hace relativamente poco tiempo cuando la información contextual ha alcanzado un papel importante en el campo de los sistemas de recomendación (Adomavicius et al., 2011b).

Herlocker y Konstan (2001) plantean utilizar un algoritmo de recomendación que extraiga cuál es la tarea que va a realizar el usuario y con esa información recomendar productos que le permitan realizar esa tarea. La

manera de identificar la tarea tiene dos versiones, una explícita, en la que el usuario proporciona esta información, y otra más refinada en la que el sistema extrae la tarea a partir del comportamiento observado en el usuario.

Posteriormente vendrían algunos trabajos centrados en identificar qué información adicional es la que se puede introducir en una recomendación y así establecer una clasificación para el contexto. Por ejemplo, Ranganathan y Campbell (2003) definen siete categorías: física (lugar, hora), ambiental (tiempo, luz, nivel de ruido), informativa (resultados deportivos, noticias), personal (estado de ánimo, salud, actividades), social (actividades en grupo, quién más está en la misma sala), aplicación (correo electrónico, sitios web) y sistema (tráfico de la red, estado de las impresoras). Por su parte Fling (2009) reduce el número de tipos a cuatro: físico, social, medio de interacción y contexto modal.

En el trabajo de Lee y Lee (2007) se describe un recomendador de música que utiliza la información demográfica y los patrones de comportamiento del usuario junto con el contexto del mismo para realizar las recomendaciones. A partir del contexto específico de un determinado usuario el sistema le recomienda aquellas canciones más escuchadas por usuarios similares en un contexto similar. El contexto está definido principalmente por cuestiones ambientales: fecha, región, estación del año, mes, día de la semana, tiempo, temperatura media del día, temperatura máxima del día, temperatura mínima del día. En (Setten et al., 2004) se propone un sistema de recomendación de guías turísticas para aplicaciones móviles llamado COMPASS. En el proceso de recomendación combinan distintos tipos de información contextual. Primeramente la información geográfica del usuario –dónde se encuentra– se utiliza para localizar servicios cercanos a él. A continuación utilizan el posible interés sobre ese recurso para el usuario con el fin de asignarle una valoración y decidir si ese recurso será recomendado o no al usuario. Este interés se determina a partir del resto de información contextual: la hora, el tiempo, la lista de la compra del usuario y la agenda del usuario.

A partir de las definiciones de contexto que mencionamos más arriba podemos intuir que la línea que separa el modelado de usuario y el contexto es muy fina, llevando a distintas interpretaciones de lo que se entiende por modelo de usuario y por contexto. De ahí que existan algunos trabajos que utilizan el contexto como medio para introducir un modelo de usuario que mejore la personalización en el sistema de recomendación. Por ejemplo Anand y Mobasher (2007) proponen utilizar un modelo de usuario que distinga entre la memoria a corto y largo plazo. Este modelo de usuario está basado en el modelo de memoria humana propuesto por Atkinson y Shiffrin (1968). En la memoria a corto plazo se almacenan las preferencias de usuario. En la memoria a largo plazo se almacena esa misma información pero para todos los usuarios del sistema. En el proceso de recomendación se generan indicaciones contextuales extraídas de la memoria a corto plazo y se utilizan

para recuperar de la memoria a largo plazo elementos relevantes que después serán utilizados en la generación de las recomendaciones.

Una alternativa interesante para incluir información sobre el contexto en los sistemas de recomendación es modelar el dominio a través de ontologías. Las ontologías pueden servir como elementos clave en los recomendadores sensibles al contexto ya que éstas permiten incluir conocimiento adicional sobre el dominio que puede ser comprendido, utilizado y compartido por distintos sistemas. Como ejemplo tenemos el trabajo propuesto por Kim y Kwon (2007). En este trabajo proponen un sistema de recomendación para realizar compras en un supermercado. Definen cuatro ontologías que serán encargadas de modelar el dominio y el contexto sobre el mismo. Existe una ontología de productos, organizada por categorías de productos; también tiene una ontología de localización que modela el conocimiento relativo a la colocación de los productos en el supermercado, una ontología con el histórico de compras del usuario y finalmente una ontología que modela las preferencias del usuario. Con el conocimiento almacenado en las ontologías de histórico de compras y preferencias de usuario se genera una recomendación a partir del conjunto de productos más comprados por el usuario. Después adapta el nivel de información que se le muestra al usuario para un determinado producto, haciendo uso de las ontologías de productos y localización (si está cerca del producto en cuestión le mostrará más información porque entiende que el interés del usuario por el mismo es mayor).

Loizou y Dasmahapatra (2006) demuestran que la inclusión del contexto en un sistema de recomendación a través del uso de una ontología es una buena manera de obtener recomendaciones más precisas. Para llegar a esta conclusión desarrollan un recomendador de música que como información contextual utiliza el último comportamiento del usuario cuando inició sesión en el sistema. Esta información está relacionada sobre todo con los gustos musicales del usuario bien a través de restricciones inferidas a partir de la observación de las preferencias del usuario (tipo de música que el sistema entiende que no le gusta) o bien a través de restricciones explícitamente definidas por el usuario (estilo musical que el usuario explícitamente ha dicho que no le gusta) y también se tienen en cuenta las tendencias globales (lo que más se está escuchando en el momento actual). Toda esta información la utiliza el sistema de recomendación –basado en el modelo del espacio vectorial– para generar las recomendaciones.

Laliwala et al. (2006) desarrollan un sistema de recomendación semántico para servicios de información agrícola. Estos sistemas utilizan la información disponible sobre el tiempo, el tipo de tierra, condiciones climáticas, etc. y ayudan en la toma de decisiones sobre qué productos cultivar y otras decisiones relacionadas con la agricultura.

Woerndl et al. (2007) proponen un sistema que recomienda aquellas aplicaciones móviles que un usuario puede instalar en su dispositivo a partir de

las aplicaciones que otros usuarios instalaron en un contexto similar –lugar en el que se encuentra el usuario. El algoritmo desarrollado primero determina el contexto del usuario actual –localización física–, después extrae la lista de aplicaciones que otros usuarios en esa misma localización tienen instaladas en su dispositivo y finalmente ordena dicha lista por el número de valoraciones positivas de cada aplicación.

2.2.3. Estrategias alternativas de interacción usuario recomendador

Tradicionalmente los sistemas de recomendación han empleado estrategias de interacción usuario-recomendador con un enfoque reactivo, esto es, el sistema reacciona ante la formulación de una consulta por parte del usuario. En muchas ocasiones, la formulación de una consulta puede suponer un esfuerzo para aquellos usuarios que no controlan el dominio o que no tienen suficiente conocimiento del mismo para realizar dicha consulta. Para aliviar este esfuerzo surgen estrategias de interacción alternativas a los enfoques reactivos, que buscan aligerar la carga de trabajo del usuario en la búsqueda del producto que mejor se adapta a sus necesidades (McGinty y Reilly, 2011; Pu et al., 2011). Estas estrategias de interacción pueden estar asociadas a un comportamiento proactivo del recomendador, es decir, el recomendador decide ofrecer recomendaciones aunque no exista una petición explícita (Bulander et al., 2005; Mahmoud, 2006; Puerta Melguizo et al., 2007; Sae-Ueng et al., 2008).

Existen alternativas intermedias entre los enfoques reactivos y proactivos que plantean el proceso de recomendación como un proceso iterativo (una *conversación*) en el que el usuario puede ir refinando sus requisitos hasta obtener productos adecuados para él y así aligerar la carga de trabajo del usuario. Básicamente, existen dos estrategias de *conversación*: navegación por propuesta o navegación por pregunta. En la navegación por propuesta, un conjunto de productos son sugeridos al usuario y el sistema obtiene realimentación del usuario con la que refinar sus requisitos. Este tipo de navegación puede ir asociado a un comportamiento proactivo en el recomendador. En este caso, el recomendador debe realizar implícitamente una petición y generar la consecuente recomendación. Por su parte, la navegación por preguntas recoge los requisitos del usuario a partir de la formulación de un conjunto de preguntas cuidadosamente seleccionadas en forma y en tiempo. La respuesta que el usuario da a cada pregunta permite ir construyendo la petición al recomendador, a partir de la cual éste generará una recomendación.

En los recomendadores de navegación por propuesta es muy importante recoger realimentación por parte del usuario para así mejorar las recomendaciones tanto a nivel de producto como en tiempo y forma. De manera general, existen tres formas de proporcionar esta realimentación:

- En forma de valoraciones, como hacen en (Smyth y Cotter, 1999; Smyth y Keane, 1998; Burke et al., 1996). Se trata de involucrar al usuario solicitando valoraciones para cada uno de los productos recuperados.
- Basado en críticas (McGinty y Smyth, 2005). En este método el usuario expresa sus restricciones en alguna de las características como se hace en el sistema Entree (Burke et al., 1996) ya presentado en este capítulo.
- Basado en preferencias, como hacen en (Smyth y Cotter, 2001; Kim et al., 2002; Yudelso y Brusilovsky, 2005). En este modelo un usuario expresa su preferencia por un producto frente a otros.

El método de navegación por propuesta se ha convertido en uno de los más populares entre los sistemas de recomendación que utilizan una estrategia de interacción conversacional. La razón principal es porque este modelo consigue un equilibrio aceptable entre el esfuerzo que un usuario debe emplear al proporcionar realimentación y la información valiosa que éste es capaz de aportar (McGinty y Reilly, 2011). Además este método ayuda a los usuarios pocos motivados para especificar completamente sus preferencias sin saber de antemano cuáles van a ser los beneficios de hacerlo (Weld et al., 2003). En la Sección 2.1.3 ya presentamos la familia de sistemas Find-Me (Burke et al., 1996) que fueron de los primeros recomendadores en incorporar un modelo de navegación basado en propuesta utilizando críticas sobre las características de los productos. Más recientemente, Pu et al. (2008) proponen un modelo de navegación basado en críticas y una serie de herramientas que facilitan la adopción de dicho método con el fin de obtener recomendaciones personalizadas. Han desarrollado una gran variedad de estudios que validan la eficacia del método en distintos dominios: organizador de viajes, búsqueda de casas, recomendadores para productos electrónicos, ordenadores, tablets y cámaras digitales (Pu y Chen, 2009). Otros ejemplos de recomendadores basados en navegación por propuesta son: Apt Decision (Shearin y Lieberman, 2001), SmartClient (Pu y Faltings, 2002, 2004), y un asistente de viajes (Linden et al., 1997).

El método de navegación por pregunta es, sin duda, una forma bastante directa de elaboración de los requisitos de un usuario y puede conducir a diálogos muy eficaces. Adaptive Place Advisor (Thompson et al., 2004), citado anteriormente en este capítulo, emplea este tipo de interacción. Este sistema sigue una aproximación que denominan *frame-based* (Bobrow, 1986): las preguntas pueden provenir del usuario o del sistema, lo cual añade complejidad al manejo de la entrada del usuario que puede realizar las preguntas que desee mediante la introducción de texto libre. El diálogo entre el recomendador y el usuario está guiado por una máquina de estados. En cada paso de diálogo el sistema actualizará las variables de estado, haciendo que pase de un estado a otro.

Una de las cuestiones clave que debe abordarse en la navegación por pregunta se refiere a la cuestión de qué tipos de preguntas se deben hacer en cada paso de la conversación. El orden y número de preguntas pueden tener un impacto significativo en la facilidad de uso y esfuerzo por parte del usuario. Doyle y Cunningham (2000) fueron de los primeros que obtuvieron resultados experimentales en este tema dentro del área del razonamiento basado en casos, mediante la evaluación de diferentes criterios de selección. Entre ellos se incluía un método basado en entropía, el cual evaluaba la ganancia de información de unas características dadas y se inspiraba el método de construcción de árboles de decisión.

Schmitt (2002) propuso una aproximación alternativa que fue especialmente hecha a medida para recomendadores en el ámbito del comercio electrónico. En lugar de utilizar la ganancia de información basada en la entropía, este enfoque consideraba la influencia que una determinada característica puede tener sobre la similitud de productos.

También existen otros trabajos que apuestan por introducir un modelo de navegación mixto. Este modelo propone modelos de interacción entre el usuario y el recomendador más flexibles (Bridge, 2002), combinando distintos tipos de navegación por propuesta y/o navegación por pregunta. Un ejemplo de sistema que utiliza esta iniciativa mixta es ExpertClerk (Shimazu, 2002). Dicho sistema implementa una combinación de navegación por pregunta y navegación por propuesta. Si bien es una alternativa interesante, conviene tener en cuenta que los sistemas que emplean iniciativa mixta necesitan mantener un modelo mucho más sofisticado que tenga información sobre el estado en el que se encuentra la conversación y cada uno de sus integrantes, y además necesita tener definido un buen protocolo de comunicación que facilite el intercambio de información entre usuario y sistema.

2.3. Sistemas de recomendación en el ámbito educativo

La introducción de las tecnologías de la información y las comunicaciones en educación supuso una regeneración de los modelos de enseñanza y aprendizaje, propiciando la aparición de nuevas herramientas que han ayudado a profesores y alumnos. Los primeros han encontrado la manera de poder reutilizar contenidos creados por otros compañeros así como crear los suyos y ponerlos a disposición de la comunidad educativa, en lo que se conoce como repositorios de recursos educativos. Para los segundos ha supuesto acceso a una gran cantidad y variedad de recursos para el aprendizaje sobre un determinado tema. También han tomado voz en su proceso de aprendizaje compartiendo aquellos recursos que más útiles les han resultado y valorando la calidad de los mismos.

Como vimos en la primera parte de este capítulo, los sistemas de re-

comendación ayudan al usuario en la selección de información donde las posibilidades de elección son muchas y variadas. En este sentido, los recomendadores son buenos candidatos a ser incorporados en los repositorios de recursos educativos con el fin de proporcionar al usuario recursos que se adapten a sus necesidades.

En esta sección haremos un repaso por lo que son los repositorios de recursos educativos, cómo son los recursos que albergan y las principales características. Acompañaremos con algunos casos concretos de repositorios de recursos educativos. A continuación se presentará cómo está siendo el trasvase de los sistemas de recomendación al ámbito educativo, analizando algunos de los sistemas que encontramos en la literatura así como el nivel en el que se abordan las líneas de investigación que más nos interesan y que señalamos en la Sección 2.2.

2.3.1. Repositorios de recursos educativos

El desarrollo de recursos educativos ha dado lugar a la formalización de un nuevo concepto: los objetos de aprendizaje. El objetivo de un objeto de aprendizaje es proporcionar un modelo de desarrollo modular basado en estándares que permite la flexibilidad del objeto, la independencia de la plataforma y la reutilización de los contenidos de aprendizaje. Existen distintas definiciones formales sobre lo que es un objeto de aprendizaje (L'Allier James, 1997; Wiley, 2000; IEEE, 15 July 2002) pero todas coinciden en señalar que los objetos de aprendizaje son recursos reutilizables y que sirven de soporte para el aprendizaje.

Estos recursos gozan de las siguientes características:

- **Accesibilidad:** un objeto de aprendizaje debe ser accesible. Esto quiere decir que los objetos de aprendizaje se marcan con metadatos y así pueden ser almacenados y referenciados desde una base de datos.
- **Reusabilidad:** un objeto de aprendizaje debe tener capacidad de ser reutilizado, es decir, una vez creado debería poder ser utilizado en diferentes contextos.
- **Interoperabilidad:** un objeto de aprendizaje debe ser independiente del gestor de conocimiento y del modo de presentación.

Cabe esperar que los objetos de aprendizaje se almacenen en algún lugar, y ya que buscamos que éstos sean reutilizables, podemos decir que estos lugares deberían ser globales. Con este propósito nacieron los repositorios de objetos de aprendizaje.

Los repositorios de objetos de aprendizaje permiten una localización común de los recursos y la reutilización de los mismos. El estándar IMS define un repositorio de objetos de aprendizaje como “cualquier colección de recursos que son accesibles a través de una red sin el conocimiento previo de

la estructura de la colección”. Pueden almacenar contenidos, metadatos o ambos (IMS Global Learning Consortium). McGreal (2008) realizó una revisión con la que concluyó que el estado de los repositorios de objetos de aprendizaje era funcional y en crecimiento pero también dejó entrever que la interoperabilidad de los objetos de aprendizaje almacenados en diferentes repositorios aún no se ha logrado.

Estos repositorios proporcionan las características deseables de cualquier biblioteca digital: mecanismos para compartir, editar y utilizar los recursos digitales. Algunos ejemplos de estos repositorios son los siguientes.

- MERLOT³ (Multimedia Educational Resource for Learning and Online Teaching) (Schell y Burns, 2002) es el repositorio de objetos de aprendizaje más conocido. Desde su creación es el que está marcando la pauta en el desarrollo de repositorios de objetos de aprendizaje. Es un repositorio centralizado que contiene sólo los metadatos de los recursos y apunta a los recursos que se encuentran ubicados en sitios remotos. Dispone de mecanismos de búsqueda de objetos de aprendizaje a partir de palabras clave como título, autor, plataforma para la que está disponible, etc. Está abierto a que cualquier persona se inscriba y se haga miembro disfrutando de todas las características del repositorio de manera gratuita.
- ARIADNE⁴ (Alliance of Remote Instructional Authoring & Distribution Networks for Europe) (Duval et al., 2001) es un proyecto que proporciona acceso a diversos repositorios de objetos de aprendizaje distribuidos por distintos países. Según su propia definición esta infraestructura cumple una doble finalidad: en primer lugar, alberga los repositorios de las instituciones colaboradoras que utilizan las herramientas ARIADNE con el fin de mantener sus repositorios; en segundo lugar, extrae y almacena localmente los metadatos de los repositorios federados que se alojan en otros lugares. Dispone de un mecanismo de búsqueda de objetos a partir de palabras clave, del repositorio de origen, idioma, formato, etc. También integra en los resultados contenidos obtenidos de páginas web de relevancia como Wikipedia, Google Books, Slide Share o Scribd.
- MACE⁵ (Metadata for Architectural Contents in Europe) (Stefaner et al., 2007) es una iniciativa Europea para integrar diversos repositorios de objetos de aprendizaje –distribuidos por distintos países– sobre temas de arquitectura en un único repositorio centralizado. El sistema propuesto cuenta con una arquitectura que permite el acceso

³<http://www.merlot.org>

⁴<http://www.ariadne-eu.org>

⁵<http://www.mace-project.eu>

a los recursos a través de mecanismos de búsqueda por categorías de productos, clasificación, autores, etc.

- POOL⁶ (Portals for Online Objects in Learning) es un consorcio de organizaciones educativas privadas y públicas que pretende crear un gran repositorio distribuido de objetos de aprendizaje, desarrollando y distribuyendo herramientas para crear repositorios conectados.
- SMETE⁷ (Science, Mathematics, Engineering and Technology Education) es un repositorio de objetos distribuidos que engloba recursos sobre las áreas de ciencias, matemáticas e ingeniería. Los recursos que se pueden encontrar están dirigidos a estudiantes y profesores universitarios. Aunque el proyecto terminó en 2010 se pueden seguir utilizando los recursos. Dispone de mecanismos de búsqueda mediante áreas de conocimiento o por palabras clave.
- Proyecto Agrega⁸, es una plataforma de objetos digitales educativos a nivel nacional creada en el año 2009. Actualmente el proyecto se ha reconvertido a Agrega2⁹ pero los objetos siguen accesibles. Contiene cerca de 100.000 objetos de aprendizaje. Los contenidos educativos que se pueden encontrar están organizados de acuerdo al currículo de las enseñanzas de niveles anteriores a la universidad. Dispone de mecanismos de búsqueda por materias, idioma, palabras clave, etc.

En la mayoría de las ocasiones estos repositorios de objetos de aprendizaje contienen tal cantidad de recursos que encontrar el que realmente se necesita se hace muy dificultoso, presentando un problema de sobrecarga de información para el usuario. Como se puede extraer de las descripciones anteriores, la mayoría de los repositorios de objetos de aprendizaje sólo ofrecen a los usuarios funciones de búsquedas muy simples (a partir de palabras clave, título, autor). Los resultados de estas búsquedas raramente serán recursos que se adapten a las necesidades y/o características del estudiante.

La actual incapacidad para localizar de manera efectiva recursos que se adapten a las necesidades de los usuarios está inhibiendo una adopción mayor de los repositorios de aprendizaje y un mayor impacto en el proceso de enseñanza-aprendizaje (Yergler, 2010; Dichev y Dicheva, 2012). Por eso es importante desarrollar mecanismos y técnicas que doten a los repositorios de objetos de aprendizaje de acceso personalizado, de tal manera que los recursos que obtenga un usuario estén adaptados a sus objetivos, preferencias y necesidades.

⁶<http://www.learningpool.com/>

⁷<http://www.smete.org/smete/>

⁸<http://www.proyectoagrega.es>

⁹<http://agrega.educacion.es>

2.3.2. Recomendadores en educación

Como ya dejamos ver en el apartado anterior, el desarrollo de nuevas técnicas que permiten compartir la información y los recursos educativos ha concluido en que nos encontramos con grandes almacenes de conocimiento a los que resulta difícil acceder de una manera personalizada y en los que resulta complicado que un usuario encuentre aquellos recursos más adecuados a su aprendizaje. Parece claro que nos encontramos ante un reto: dotar de buenas herramientas de localización a los repositorios de recursos educativos que tengan en cuenta las peculiaridades del dominio así como las características del usuario. Los sistemas de recomendación son unos claros candidatos a incorporarse como herramienta fundamental en los repositorios de recursos educativos y en los sistemas de gestión del aprendizaje, de manera que ayuden al usuario a encontrar los recursos que mejor se adapten a sus necesidades.

En este sentido, en la última década han surgido diversos trabajos que han intentado abordar el problema de la sobrecarga de información en los repositorios de objetos de aprendizaje proponiendo herramientas de recomendación para adaptar sus contenidos al usuario consiguiendo la personalización de los sistemas. De acuerdo a la clasificación propuesta por Manouselis et al. (2011a), podemos decir que existen tres tareas principales que puede llevar a cabo un recomendador para dotar de personalización a un repositorio de objetos de aprendizaje: recomendación de recursos, recomendación de caminos de aprendizaje dentro del repositorio y recomendación de otros usuarios que sirvan de ayuda en el proceso de aprendizaje. En la literatura encontramos estos tres tipos de trabajos, aunque frecuentemente algunos sistemas realizan varias de estas tareas a la vez. Concretamente la tarea de recomendar usuarios suele estar vinculada a la recomendación de recursos, por lo que en la revisión realizada algunos sistemas que realizan principalmente una de las dos primeras tareas además añaden la recomendación de usuarios.

Comenzaremos esta revisión presentando principalmente trabajos centrados en la primera tarea –recomendar recursos. El primer trabajo que citamos presenta una introducción al filtrado colaborativo en el dominio de la educación a través del sistema *Altered Vista* (Recker y Walker, 2003), siendo éste uno de los primeros trabajos en utilizar esta técnica en el ámbito educativo. *Altered Vista* recomienda páginas web con recursos educativos a un estudiante. Funciona como un recomendador colaborativo que utiliza el algoritmo de los vecinos más cercanos para construir el vecindario y después elabora predicciones sobre los recursos. Para calcular la similitud entre usuarios, éstos han tenido que valorar al menos dos recursos iguales y tener al menos una correlación de 0,5. Los recursos son introducidos en la base de conocimiento del recomendador por los propios usuarios. Éstos deben introducir una pequeña descripción de cada recurso indicando palabras clave, uso educativo y una pequeña revisión. En un trabajo posterior (Walker et al., 2004) encon-

tramos un estudio inicial en el que los autores incluyen la recomendación de personas (otros estudiantes) con intereses similares al estudiante actual que pueden servir al propio alumno como colaboradores y/o como ayuda en el aprendizaje.

Otro trabajo que también emplea la estrategia de filtrado colaborativo es CoFIND (Dron et al., 2000b,a). CoFIND es un repositorio web creado y mantenido por los propios estudiantes, que emplea filtrado colaborativo para localizar recursos educativos en función de unas medidas de calidad asignadas a cada recurso por la comunidad educativa. Algunas de las calidades existentes en el sistema son: útil, interesante, informativo, fidedigno y amplia cobertura. Las calidades de cada recurso son generadas a partir de lo que dicen los usuarios y de la estimación del sistema. Los recursos son añadidos por cada estudiante en forma de dirección web al repositorio junto con las calidades que el usuario cree que dicho recurso cumple. Además el sistema propone a cada usuario que revise el conjunto de calidades que está asociado a un recurso, indicando cuáles de ellas cree que son ciertas. La recomendación se hace en base a la frecuencia de uso de cada calidad. Es decir, a un usuario, cuando entra en el sistema, le aparece una lista de las calidades más utilizadas y un cuadro de búsqueda. La búsqueda de recursos se hace en función de la calidad que seleccione. La lista de recomendaciones contiene cinco recursos y está ordenada en función del número de votos de la calidad seleccionada que tengan los recursos.

CYCLADES (Avancini y Straccia, 2005) es un sistema donde los usuarios buscan, acceden y evalúan recursos educativos disponibles en los repositorios federados a la Open Archives Initiative¹⁰. En CYCLADES podemos encontrar dos tareas de recomendación, por un lado la recomendación de recursos y, por otro, la recomendación de usuarios y/o comunidades que pueden ser de interés para un usuario concreto. Para ambas tareas la estrategia de recomendación propuesta sigue un enfoque colaborativo identificando los temas de los recursos y los intereses de los usuarios y analizando las valoraciones asignadas por cada usuario.

En (Tang y McCalla, 2003) detallan un sistema de aprendizaje en línea adaptativo que evoluciona en función del usuario y de los contenidos que éste utiliza. El sistema de aprendizaje contiene un número fijo de recursos sobre un determinado tema. A medida que los usuarios lo utilizan, el propio sistema navega por la web buscando nuevos recursos que puedan ser de utilidad al usuario, adaptando así el sistema a las necesidades del usuario. Cuando el sistema encuentra un nuevo recurso el usuario debe valorarlo para decidir si se añade al sistema o se descarta. La recomendación y búsqueda de nuevos recursos se realiza a partir del historial de navegación del usuario recomendando así recursos similares a los ya utilizados.

Tsai et al. (2006) proponen un modelo de recomendación híbrido de obje-

¹⁰<http://www.openarchives.org>

tos de aprendizaje que utiliza dos algoritmos, uno basado en las preferencias del usuario y otro basado en los intereses de los usuarios similares. Las preferencias del usuario se obtienen a partir del historial de navegación del usuario obteniendo utilidades mayores aquellos objetos de aprendizaje que coinciden en alguna característica con los ya visitados por el usuario. Para representar los objetos de aprendizaje y los contenidos del curso los autores proponen el uso de ontologías que permitan así extraer cuáles son los objetos de aprendizaje más adecuados en cada momento de estudio. Este trabajo sólo presenta un modelo teórico que más tarde será retomado en (Wang et al., 2007) para la implementación de un sistema de recomendación de objetos de aprendizaje sobre el lenguaje de programación Java. Existen otros trabajos que también utilizan ontologías para añadir información semántica que facilite la recuperación de recursos. Más concretamente, el trabajo presentado en (Piedra et al., 2010, 2011) propone una búsqueda social semántica que encuentra aquellos objetos de aprendizaje que cumplen con los requisitos de la consulta realizada por el usuario. Estos objetos de aprendizaje se extraen a partir de una búsqueda semántica sobre el conocimiento de los objetos de aprendizaje. Otro trabajo que también hace uso de información semántica es el propuesto en (Shelton et al., 2010). En dicho trabajo proponen OER Recommender, un recomendador basado en contenido que recomienda objetos de aprendizaje a partir de la información semántica contenida en los metadatos de los mismos (OER Recommender forma parte del proyecto *Folksemantic*).

Otro trabajo reciente en recomendación de objetos de aprendizaje es el propuesto por Ghauth y Abdullah (2011) en el que los autores describen un sistema de recomendación híbrido que combina una estrategia de recomendación basada en contenido y una colaborativa. El módulo de recomendación basada en contenido utiliza el modelo del espacio vectorial para calcular la similitud entre cada par de objetos de aprendizaje y seleccionar los n más similares al que el alumno está visualizando. El módulo de recomendación colaborativa utiliza las valoraciones asignadas por los *buenos alumnos* a los objetos de aprendizaje. Estas valoraciones son utilizadas para presentar el conjunto de los n objetos de aprendizaje más similares junto con la valoración media que tiene cada recurso para los *buenos alumnos*. Para saber quién es un *buen alumno* el sistema realiza a cada estudiante un examen inicial para evaluar su conocimiento sobre la materia. Aquellos que obtienen una puntuación superior al 80 % de la nota máxima son considerados *buenos alumnos*.

Por su parte Zapata et al. (2011) describen una estrategia de recomendación híbrida que combina cuatro algoritmos de filtrado colaborativo. Estos cuatro algoritmos de filtrado son: filtrado por similitud en el contenido, filtrado por uso, filtrado por calidad y filtrado por similitud con la consulta. El resultado de cada uno de estos algoritmos es una puntuación para el objeto de aprendizaje en cuestión. Después existe una fase de ordenación de

los objetos, de tal manera que cada producto es ordenado de acuerdo a una combinación ponderada del valor obtenido para cada algoritmo.

Finalmente (Michlík y Bieliková, 2010) proponen un método para realizar recomendaciones personalizadas de ejercicios en un sistema educativo en el que el tiempo para realizar dichos ejercicios tiene un límite. Este sistema está pensado para conseguir en un tiempo fijo los conocimientos de un determinado curso y así pasar el examen final. Para lograr su objetivo, la estrategia de recomendación propone tareas y ejercicios que comprendan el mayor número posible de temas para que el estudiante alcance un nivel adecuado en todos en lugar de conocer a la perfección sólo un subconjunto de temas. La estrategia de recomendación utilizada se basa en recomendar aquellos objetos de aprendizaje con mejor utilidad para un determinado alumno. La utilidad de cada recurso se calcula de acuerdo a tres criterios: la pertinencia de los conceptos para el alumno, la dificultad del ejercicio y el tiempo transcurrido desde la última vez que el alumno intentó realizar el ejercicio. Asimismo, realizan un experimento en la plataforma ALEF para un curso de programación lógica. El experimento lo realizan con tres grupos: sin recomendación, recomendación con un modelo automático y recomendación con un modelo personalizado. Los resultados muestran que los alumnos que utilizan el sistema con recomendación obtienen mejores resultados en el examen final que los que no, aunque, como los propios autores indican, deben mejorar el trabajo porque en su primer experimento se muestra que el grupo que recibió ejercicios aleatorios (sin recomendación) obtuvo prácticamente el mismo número de estudiantes que superaron el examen que los que recibieron recomendaciones con un modelo personalizado.

En cuanto al segundo tipo de trabajos que comentamos al comienzo de la sección –los que se centran principalmente en recomendar actividades o secuencias de aprendizaje– podemos comenzar con el descrito en (Zaiane, 2002), donde el autor utiliza técnicas de aprendizaje máquina sobre el historial de navegación de los alumnos en un sistema educativo web con el fin de obtener patrones de comportamiento. Lo que se busca es poder realizar después recomendaciones sobre las actividades en base a esos patrones. Posteriormente, con ayuda de reglas de asociación, construye un sistema de recomendación que, a partir del patrón de comportamiento del alumno actual, recomienda qué actividad podría hacer a continuación.

Otra manera de adaptar la navegación es la propuesta por Hummel et al. (2007). En este trabajo presentan un sistema de recomendación que propone al alumno un camino de aprendizaje para llevar a cabo su estudio. El camino de aprendizaje está compuesto por distintas actividades que el alumno debe ir completando dentro del sistema educativo (escribir un mensaje en un foro, leer una noticia, etc). Las recomendaciones se realizan en base al perfil del alumno y a los objetivos del curso que tiene que completar. Además proponen un experimento con alumnos reales cuyos resultados detallan en (Drachsler

et al., 2009). La principal conclusión de esta evaluación es que muestran que las recomendaciones tienen un efecto positivo en el aprendizaje de los alumnos reflejado en la eficiencia, es decir, en el tiempo que el alumno tarda en completar la actividad propuesta en un objeto de aprendizaje.

El último trabajo que vamos a presentar en esta línea es el propuesto por Santos y Boticario (2008a,b). Este trabajo presenta recomendaciones personalizadas sobre las actividades que un determinado alumno debería llevar a cabo en un sistema gestor de recursos educativos. En la primera parte del trabajo Santos y Boticario (2008a) analizan cuáles son las actividades más adecuadas a los alumnos en función de su nivel. Esto lo realizan con un experimento con expertos en el área que evalúan cada actividad, para después integrar las recomendaciones en un sistema web educativo (Santos y Boticario, 2008b). Para realizar las recomendaciones, los autores primeramente definen un conjunto de 25 posibles actividades que se pueden realizar dentro del curso (ver un mensaje del foro, escribir un correo al tutor, leer la ayuda, etc). Después definen las reglas para ejecutar cada actividad. Las recomendaciones están clasificadas por su origen (quién propone la recomendación), social (usuarios similares han realizado una determinada actividad), contenido (el contenido de la recomendación encaja con las preferencias del usuario), propuesta por el tutor (la recomendación ha sido originada por el tutor del alumno en el curso) o diseñador del curso (la recomendación ha sido propuesta por el diseñador del curso). A su vez, estas recomendaciones tienen unas condiciones de aplicación y cada una está englobada en un tipo. Con toda esta información relativa a una recomendación se construyen reglas de asociación que producirán recomendaciones en tiempo real.

Finalmente, señalar que la tendencia de incluir contexto en el proceso de recomendación también ha sido recogida por la comunidad educativa creciendo en los últimos años el número de recomendadores en educación que incorporan el contexto a su sistema.

En (Verbert et al., 2012), un trabajo muy reciente que se centra en analizar las características del contexto en el ámbito educativo, los autores proponen un marco común que unifica las diversas definiciones de contexto e intenta definir las categorías del contexto y los datos dentro de estas categorías. Como resultado, introducen una clasificación del contexto en categorías a partir del análisis de las definiciones existentes. Estas categorías son: informática (características de hardware y software), localización, hora, condiciones físicas (temperatura, tiempo, espacio ruidoso), actividad (tareas, objetivos o acciones que debe realizar el usuario), recurso (características relevantes de los recursos), usuario (modelo de usuario) y relaciones sociales. A partir de estas características proponen organizar los trabajos existentes, aunque por lo general los trabajos suelen combinar más de un tipo de contexto.

En cuanto a sistemas concretos que usan información contextual pasamos

a presentar algunos recientes. Un primer ejemplo es el trabajo propuesto por Zhao et al. (2008) que presenta un recomendador para dispositivos móviles en el que la recomendación utiliza información contextual como la localización, hora, rendimiento de la red, tipo de red y tipo de dispositivo móvil para recomendar aquellos recursos que mejor se adaptan al dispositivo, al software disponible y a las restricciones de la red. Esto lo hacen mediante un algoritmo adaptativo que opera en tres etapas: recomendación de contenido, negociación y transformación del contenido. El esquema de recomendación adaptativo propuesto lo han ejemplificado con un sistema de prueba para el aprendizaje de las matemáticas en educación primaria.

En (Lehsten et al., 2010) introducen un sistema que adapta los contenidos de un LMS (Learning Management System) a una aplicación para dispositivos móviles. Su propuesta se basa en utilizar la localización del alumno para ofrecerle servicios en el campus universitario y recursos educativos adaptados al lugar físico en el que se encuentra. En este caso, entienden por contexto la descripción de la situación actual del alumno, que se caracteriza por la identidad del mismo, el dispositivo que está utilizando para acceder al LMS, los cursos en los que participa, dónde está en este momento y cualquier otra información relevante relacionada con estos aspectos. Con toda esta información contextual el sistema puede, por ejemplo, recomendar productos cercanos al alumno –un libro en la biblioteca o una exposición en un museo. Por otra parte, el sistema también es capaz de reaccionar a limitaciones en el ancho de banda y recomendar así productos en otros formatos multimedia.

El siguiente ejemplo combina la hora y las condiciones físicas para realizar la recomendación. Yau y Joy (2007) presentan un entorno de aprendizaje adaptativo que utiliza como contexto el horario del alumno (tiempo que dispone para el estudio) y la localización (lugar físico y condiciones de ruido). Estas características, unidas al estilo de aprendizaje del alumno, serán empleadas por el sistema para proponer actividades de aprendizaje adaptadas a estas características.

Existen también otros trabajos que como parte de la información contextual tienen en cuenta la actividad que está realizando el alumno para sugerir recursos que puedan ser relevantes para dicha tarea. Un ejemplo es el sistema propuesto en (Stern et al., 2010) en el que identifican el tema al que pertenece una actividad para así recomendar recursos que se adapten a dicho tema. Por su parte, Teng et al. (2011) proponen un sistema que ayuda en el aprendizaje de un idioma identificando la actividad que está realizando un estudiante para enlazar con otros recursos que le podrían interesar. Por ejemplo, recomiendan recursos digitales relacionados con el fragmento de texto que el estudiante está leyendo.

El último ejemplo que vamos a presentar es el trabajo propuesto por (Jeremic et al., 2009). En este trabajo presentan DEPTH, una herramienta para facilitar el aprendizaje de los patrones de diseño en programación. La

recomendación la realizan teniendo en cuenta el contexto social y de usuario. El primero lo utilizan de manera que recomiendan recursos que otros alumnos similares han utilizado o recomiendan colaboradores para realizar prácticas. El contexto del usuario se utiliza para conocer qué recursos ha utilizado el alumno y cuáles (en función de la similitud de contenido y etiquetas) pueden ser relevantes para él.

2.4. Resumen y conclusiones

En este capítulo se ha realizado una revisión del estado del arte de los sistemas de recomendación señalando las principales estrategias existentes así como algunos sistemas que las ejemplifican y algunas líneas de investigación abiertas. Posteriormente se ha realizado una introducción a los repositorios de objetos de aprendizaje y una revisión de los trabajos de recomendación en educación más relevantes, para comprender cuál está siendo la aplicación de los sistemas de recomendación en este dominio.

En el caso de los sistemas de recomendación se ha visto que existen distintas alternativas de diseño de los mismos en función de dónde proviene la información que va a utilizar –colaborativos, basados en contenido y basados en conocimiento. También están tomando protagonismo los sistemas híbridos que surgen como combinación de algunos de los anteriores. Una de las técnicas más empleada en los sistemas de recomendación es la conocida como filtrado colaborativo. Esta técnica necesita de la existencia de valoraciones para realizar la recomendación por lo que no es capaz de producir recomendaciones para usuarios nuevos en el sistema ni para productos que acaban de ser incluidos en el catálogo de productos a recomendar. Por su parte los recomendadores basados en contenido utilizan las descripciones de los productos disponibles junto con las preferencias del usuario sobre esos productos para generar recomendaciones. En general usan técnicas conocidas de recuperación de información para predecir las valoraciones de los productos en función de las descripciones de los productos ya valorados por los usuarios. La principal debilidad de estos sistemas es la sobreespecialización de los resultados obtenidos. Por último, los recomendadores basados en conocimiento utilizan, además de la información relacionada con los productos del sistema, fuentes de información adicional para la realización de la recomendación, como las características de los productos, las preferencias del usuario y conocimiento del dominio para, en base a las restricciones impuestas por el usuario, generar una recomendación. Estas técnicas de recomendación están en auge ya que alivian muchos de los problemas detectados en las anteriores técnicas de recomendación y cada vez hay más fuentes de información disponibles sobre las que obtener información adicional de dominio, gracias a la computación en la nube (*cloud computing*) y las interfaces de programación abiertas que cada vez más páginas web proporcionan.

Hemos presentado también en este capítulo tres líneas de investigación abiertas en los sistemas de recomendación: la introducción de diversidad en las listas de recomendación, la incorporación de información contextual en el proceso de recomendación y la exploración de modelos de interacción que alivien la carga de trabajo del usuario. La introducción de diversidad se propone como un método para otorgar al usuario variedad dentro de los productos recomendados. Por su parte, la incorporación de la información contextual en los sistemas de recomendación brinda nuevas oportunidades de personalización en éstos, ya que el sistema tiene más conocimiento del usuario pudiendo adaptar mejor el conjunto de productos recomendados. Finalmente, los modelos de interacción alternativos a la formulación de una consulta directa brindan nuevas facilidades de interacción con el recomendador, haciendo que usuarios poco expertos en el dominio o con una actitud pasiva hacia el mismo obtengan recomendaciones sin suponer un esfuerzo para ellos. Estos modelos de interacción también facilitan la obtención de las preferencias del usuario por parte del sistema de una manera no intrusiva para él.

En el campo de los repositorios de objetos de aprendizaje se ha visto que en éstos suele ser habitual la carencia de mecanismos de búsqueda que permitan la localización personalizada de recursos aunque la mayoría sí cuentan con búsqueda por texto libre o por categorías de recursos. Estos repositorios son un buen campo de experimentación para mostrar el potencial de las técnicas de recomendación. Como se ha podido ver, la mayoría de los recomendadores presentados en el dominio educativo utilizan técnicas de filtrado colaborativo o basadas en contenido, dejando fuera la riqueza que supone para una estrategia de recomendación utilizar todo el conocimiento sobre el dominio y/o sobre los usuarios y su contexto.

De cara al trabajo de esta tesis, la revisión realizada nos da las pautas para diseñar distintas estrategias de recomendación aplicadas a la búsqueda personalizada en repositorios de recursos educativos. Por un lado, nuestro interés se va a centrar en explorar el potencial de las técnicas de recomendación basadas en conocimiento. Al igual que otros autores, creemos que el uso de estas técnicas en un campo tan complejo como el educativo puede producir muy buenos resultados en los recomendadores desarrollados. Por otro lado, se ha comprobado que si bien la inclusión de diversidad es un tema candente en recomendación, no se saca partido de él en recomendadores en el ámbito educativo, pese a la capacidad para ofrecer al estudiante nuevas alternativas de estudio para alcanzar sus objetivos. Nuestro trabajo realizará avances en esta línea. Finalmente, los trabajos de recomendación relacionados con el uso de la información contextual en el ámbito educativo están aprovechando aspectos como la geolocalización o la tarea que el alumno está realizando, pero no explotan objetivos formativos a más largo plazo. El presente trabajo supone un paso más en el desarrollo de nuevas estrategias de recomendación

que se adapten mejor a las características de cada usuario, ya que se propone hacer uso de la información contextual de la actividad (según la clasificación propuesta por Verbert et al. (2012)) e información contextual en forma de un modelo de usuario al estilo que apuntaban en Anand y Mobasher (2007). En este sentido, coincidimos con el trabajo presentado en Anand y Mobasher (2007) en que el uso del contexto como medio para introducir un modelo de usuario puede ser un buen método para mejorar la calidad de las recomendaciones. Y, como proponen otros autores, la incorporación del contexto se hará gracias a modelar el dominio a través de ontologías.

Finalmente, hemos visto que los trabajos existentes tampoco se preocupan por aliviar el esfuerzo y el tiempo empleado por un estudiante en encontrar el recurso que mejor se adapte a sus conocimientos, siendo ésta una de las principales demandas que tienen los usuarios de los repositorios de contenidos educativos (Dichev y Dicheva, 2012). Por este motivo, apostamos por explorar un modelo de navegación por propuesta que sugiera de manera proactiva al estudiante un primer conjunto de objetos de aprendizaje adaptados a sus necesidades, y que permita de manera sencilla la obtención de realimentación del usuario. En resumen, el trabajo que se presenta en esta tesis intenta avanzar en mejorar la adaptación al contexto del estudiante, la incorporación de diversidad en los recursos recomendados y la exploración de estrategias de interacción con un comportamiento proactivo complementando con facilidades de navegación por propuesta, aspectos todos ellos considerados muy deseables por los usuarios de los repositorios de objetos de aprendizaje.

Capítulo 3

Estrategias de recomendación para repositorios de objetos de aprendizaje

La única manera de hacer muchas cosas a la vez es hacerlas de una en una.

Máxima 7, Publilius Syrus

En el capítulo anterior veíamos que los repositorios de objetos de aprendizaje han tomado un gran protagonismo en la introducción de las tecnologías en el ámbito educativo. La utilidad de estos repositorios se basa en la cantidad y la calidad de los recursos que albergan, pero si el repositorio no cuenta con unos buenos mecanismos de búsqueda y localización la utilidad desaparece. Por ello, tal y como concluíamos en el capítulo anterior, sería deseable que los repositorios contaran con mecanismos de recomendación de recursos que encajen en las necesidades y en el contexto del estudiante actual promoviendo su aprendizaje, que fomenten la diversidad de los recursos sugeridos y/o exploren modelos de interacción alternativos a la formulación de una consulta.

El trabajo de esta tesis se ha centrado en el desarrollo de estrategias de recomendación que permitan el acceso personalizado a los objetos de aprendizaje contenidos en repositorios educativos y que afronten los aspectos citados anteriormente. A lo largo de este capítulo presentaremos las estrategias de recomendación desarrolladas.

Comenzaremos el capítulo describiendo la estrategia de recomendación de objetos de aprendizaje que ha servido de punto de partida al presente trabajo y analizando los inconvenientes achacables a la misma (Gómez-Albarrán y Jiménez-Díaz, 2009) (Sección 3.1). En dicho trabajo se presentó una estrategia de recomendación que seguía un enfoque muy simplista basado en

la consulta que formulaba el estudiante, dando como resultado recomendaciones con una muy reducida personalización. Uno de los objetivos de este trabajo de tesis ha sido proponer soluciones alternativas que solventaran las carencias mostradas por esta primera aproximación.

A continuación, en la Sección 3.2, introduciremos las estrategias de recomendación que surgen al querer solventar las carencias detectadas en la primera estrategia. El conocimiento necesario para las nuevas estrategias de recomendación será descrito en la Sección 3.3.

En la última parte del capítulo describiremos en profundidad el funcionamiento de estas nuevas estrategias de recomendación propuestas. La primera aproximación (Sección 3.4) pretende mejorar la capacidad de adaptación al usuario explorando un modelo de personalización fuerte en el que la información contextual asociada a la actividad y al estudiante jugará un papel clave. La Sección 3.5 presenta una estrategia que, a partir de la consulta planteada por un estudiante, recomienda un conjunto de objetos de aprendizaje que cumplen los objetivos a corto plazo reflejados en la consulta, aliviando al mismo tiempo la sobreespecialización producida por los enfoques basados en similitud y manteniendo cierto nivel de personalización adicional gracias al uso de la información contextual. Por último, la tercera propuesta (Sección 3.6) promueve la interacción con el estudiante mediante un mecanismo que combina proactividad y navegación por propuesta por el espacio de las recomendaciones realizadas teniendo en cuenta la información contextual. Este modelo de interacción alivia la naturaleza reactiva de la estrategia inicial que impone que el estudiante tenga que realizar una consulta. Gracias a él, el estudiante puede realizar la localización de los objetos de aprendizaje sin necesidad de formular una consulta. Terminaremos el capítulo con las conclusiones obtenidas (Sección 3.7).

3.1. Planteamiento y debilidades de una primera estrategia de recomendación

El trabajo desarrollado en esta tesis parte de una primera aproximación presentada en (Gómez-Albarrán y Jiménez-Díaz, 2009) en la que se describía un enfoque novedoso para recomendar objetos de aprendizaje. En dicho trabajo se planteaba una estrategia de recomendación híbrida en cascada: un enfoque de recomendación reactivo, *single-shot*, basado en conocimiento, actuaba en primer lugar y sus decisiones eran refinadas a continuación por un recomendador basado en filtrado colaborativo. La estrategia de recomendación híbrida empleada localizaba, en un único paso de interacción, un conjunto relevante de objetos de aprendizaje después de que el estudiante hubiera planteado una consulta al sistema. El resultado era una lista ordenada de objetos de aprendizaje. La prioridad era para aquellos objetos de aprendizaje más similares a la consulta del estudiante y mejor valorados por

otros estudiantes. La estrategia estaba soportada por una indexación de objetos de aprendizaje basada en una ontología del dominio. Como veremos, este esquema de indexación es crucial para el recomendador basado en conocimiento. Explicaremos a continuación cómo funcionaba esta estrategia híbrida.

El recomendador basado en conocimiento que actuaba en un primer momento lo hacía en dos etapas: recuperación y valoración. La etapa de recuperación buscaba objetos de aprendizaje que satisficieran, de una manera aproximada, los objetivos de aprendizaje del estudiante. El estudiante planteaba una consulta al sistema utilizando conceptos del dominio. Esta consulta representaba sus objetivos de aprendizaje en esa sesión de estudio. La etapa de recuperación buscaba aquellos objetos de aprendizaje indexados por los conceptos de la consulta. Si no había objetos de aprendizaje que satisficieran esta condición, la recuperación exacta anterior se podía sustituir por una recuperación aproximada que se realizaba teniendo en cuenta un subconjunto de los mismos o similares conceptos a los planteados en la consulta del estudiante, para lo cual la ontología de conceptos del dominio era una fuente de conocimiento fundamental. Una vez que los objetos de aprendizaje habían sido recuperados, la etapa de valoración calculaba la relevancia (o calidad) asignada a cada objeto de aprendizaje. La relevancia de un objeto de aprendizaje L para un estudiante S se calculaba como la suma de dos elementos:

1. *La relevancia debida a los objetivos cumplidos por L .* Cuanto mayor era el número de conceptos de la consulta que L permitía aprender, mayor era el valor de la relevancia. Cuanto más similares eran los conceptos de la consulta y los que L permitía aprender, mayor era el valor de la relevancia.
2. *La relevancia debida al grado de adaptación de L al conocimiento actual de S .* El conocimiento de S estaba representado en su perfil dentro de un repositorio de perfiles de estudiantes. El objetivo era penalizar a L si incluía conceptos (diferentes de aquellos que satisfacen la consulta) que no se encontraban en el perfil de S .

Después de que el recomendador basado en conocimiento hubiera obtenido un conjunto de objetos de aprendizaje candidatos a ser recomendados, el recomendador basado en filtrado colaborativo refinaba dicho conjunto hasta obtener un subconjunto final de los mismos. Esto lo hacía combinando la calidad asignada a cada objeto de aprendizaje L por el recomendador anterior con un valor de relevancia que representaba la utilidad que había sido asignada a L por otros estudiantes con perfiles similares a S . Un repositorio de preferencias de estudiantes almacenaba las valoraciones asignadas a los objetos de aprendizaje utilizados por los estudiantes junto con el perfil del

estudiante que los valoró en el momento de utilizarlos. El peso asignado a cada valoración se asignaba de acuerdo a la similitud entre el perfil del estudiante que valoró el objeto de aprendizaje y el actual perfil de S . Cuanto más similares eran los perfiles, mayor era la relevancia asignada a la valoración.

Esta estrategia híbrida aliviaba algunos de los problemas asociados a los recomendadores basados en filtrado colaborativo (explicados en el capítulo anterior). Si un objeto de aprendizaje tenía pocas valoraciones o un estudiante era nuevo en el repositorio, el recomendador devolvía un conjunto de objetos de aprendizaje relevantes gracias a la recomendación basada en conocimiento. Por otra parte, el filtrado colaborativo basado en usuario que actuaba en segundo lugar tras el recomendador basado en conocimiento no necesitaba explorar el repositorio completo de preferencias de usuario para buscar estudiantes similares. Los perfiles escogidos eran aquellos que ya habían valorado un objeto de aprendizaje candidato (aquel sugerido por el recomendador basado en conocimiento).

Aunque a esta estrategia de recomendación se le pueden atribuir las ventajas enunciadas, también es posible achacarle los siguientes inconvenientes:

- Primero, la débil personalización proporcionada por la recomendación basada en conocimiento descrita anteriormente. Puede decirse que dicha estrategia implementa un tipo de personalización que se limita, en gran medida, a las necesidades manifestadas por el estudiante en la sesión de recuperación: se toma en cuenta los objetivos a corto plazo propuestos en forma de consulta. De esta manera, para dos estudiantes que planteen la misma consulta en una sesión se obtendrán las mismas recomendaciones en el proceso de recuperación, incluso si sus objetivos de aprendizaje a largo plazo y su destreza en el dominio difieren en gran parte. El conocimiento que del dominio tiene previamente el estudiante sólo se tiene en cuenta mínimamente en la etapa de valoración, a la hora de calcular la relevancia debida al grado de adaptación del objeto de aprendizaje al conocimiento del estudiante.
- Otra de las desventajas es el problema ya anunciado en el capítulo anterior. La estrategia de partida emplea la acumulación de relevancia individual de cada objeto de aprendizaje como mecanismo para promocionar a los objetos de aprendizaje hacia el conjunto de recomendados. Esto da lugar a que los objetos de aprendizaje mejor clasificados apenas difieran unos de otros. Consecuentemente, cuando a un estudiante no le gusta el primer objeto de aprendizaje recomendado es probable que tampoco esté satisfecho con los otros. Esto supone un problema en cualquier contexto, y en el nuestro en particular, donde parece conveniente restringir el número de recursos que se mostrarán al estudiante para así no sobrecargarle en cada sesión de estudio. El resultado puede llegar a ser que ninguno de los recursos mostrados sea satisfactorio.

- La última desventaja radica en la naturaleza reactiva de la estrategia propuesta. El estudiante debe formular una consulta en términos de los conceptos en los que está interesado. El planteamiento de la consulta puede imponer una empinada curva de aprendizaje de uso para aquellos estudiantes con escasos conocimientos. Esto además puede ser un obstáculo para estudiantes con una actitud pasiva.

Estas desventajas nos llevaron a plantear nuevas estrategias de recomendación basadas en conocimiento que pudieran sustituir a la actual y que alivien estas dificultades. Las nuevas estrategias basadas en conocimiento, de validez por sí mismas, podrían seguir siendo encajadas en una estrategia híbrida como la descrita aquí.

3.2. Afrontando debilidades mediante nuevas estrategias de recomendación basadas en conocimiento

Una vez vistas las debilidades de la primera aproximación nuestro trabajo ha ido dirigido hacia la definición de nuevas estrategias de recomendación que aliviaran los problemas planteados. Estas nuevas estrategias necesitan incluir conocimiento adicional sobre el dominio y sobre el estudiante para así poder conseguir una mayor personalización en las recomendaciones. El conocimiento del dominio permitirá, entre otras cosas, establecer unos objetivos de aprendizaje –fijados por el profesor– que todos los estudiantes deberán completar. Por otro lado, el conocimiento sobre un determinado estudiante aportará información sobre su estado cognitivo, permitiendo al recomendador encontrar aquellos objetos de aprendizaje que le permitan mejorar en su nivel de aprendizaje. Estas nuevas estrategias, cuyo nexo común se encuentra en la base rica en conocimiento empleada para hacer las recomendaciones, son las siguientes:

- La primera estrategia (Sección 3.4) pretende mejorar la capacidad de adaptación de las recomendaciones al usuario explorando un modelo de personalización fuerte. Esta estrategia se concibe como un recomendador basado en conocimiento que mantiene un modelo de operación reactivo, esto es, bajo demanda del estudiante: el estudiante deberá formular una consulta indicando qué es lo que quiere aprender en su sesión de estudio. A la hora de recomendar objetos de aprendizaje veremos que se tiene en cuenta no sólo la consulta formulada por el estudiante sino también la información contextual que nos detalla el conocimiento que el estudiante tiene del dominio, incorporando así una personalización mayor, y los objetivos de aprendizaje. Esta estrategia asigna la prioridad a aquellos objetos de aprendizaje que son más similares a la

consulta del estudiante (objetivos a corto plazo) y, al mismo tiempo, tienen una utilidad pedagógica mayor de acuerdo a la información contextual (objetivos a largo plazo). Como veremos, en nuestro caso hemos decidido aplicar una estrategia pedagógica que asigna altas utilidades pedagógicas a aquellos objetos de aprendizaje que permiten remediar carencias de conocimiento reflejadas en el perfil del estudiante.

- La segunda estrategia (Sección 3.5) se centra en aliviar el problema de la sobreespecialización explorando la incorporación de un mecanismo de diversidad inspirado en el algoritmo propuesto por (Smyth y McClave, 2001). El objetivo es recomendar un reducido número de objetos de aprendizaje donde los conjuntos de conceptos cubiertos por distintos objetos de aprendizaje sean ligeramente diferentes entre sí, pero que sean de interés para el estudiante actual. La adaptación del algoritmo de diversidad arriba mencionado nos permitirá seleccionar los objetos de aprendizaje que finalmente serán propuestos al estudiante.
- La tercera estrategia (Sección 3.6) se centra en aliviar la sobrecarga que acompaña a la naturaleza reactiva del recomendador inicial. Para ello, esta estrategia seguirá un comportamiento proactivo que propondrá a los estudiantes objetos de aprendizaje que, teniendo en cuenta la información contextual, puedan ser de su interés en una sesión de estudio sin la necesidad de formular una consulta explícita. Esta estrategia se ha diseñado teniendo en mente un esquema de navegación por propuesta. De esta manera, conseguimos captar al estudiante en un sencillo proceso conversacional que evita la realización de preguntas directas, pero que presenta recomendaciones alternativas y solicita al estudiante realimentación basada en preferencias: el estudiante mostrará predilección por una alternativa frente a las otras, haciéndole así participe del proceso de recomendación.

Como hemos señalado ya, estas tres aproximaciones seguirían encajando perfectamente en el recomendador híbrido propuesto en (Gómez-Albarrán y Jiménez-Díaz, 2009), donde la nueva recomendación basada en conocimiento dirigiría inicialmente la búsqueda de los posibles objetos de aprendizaje interesantes y la recomendación basada en filtrado colaborativo ayudaría a seleccionar los objetos de aprendizaje finalmente propuestos al estudiante. Este planteamiento se encuentra descrito en (Ruiz-Iniesta et al., 2012b), donde se presenta un modelo de gestión de repositorios de objetos de aprendizaje que proporciona a una comunidad educativa: (a) facilidades de autoría y mantenimiento que permiten incluir nuevos objetos de aprendizaje y valorarlos y (b) facilidades híbridas de recomendación. La Figura 3.1 muestra un esquema de dicho modelo.

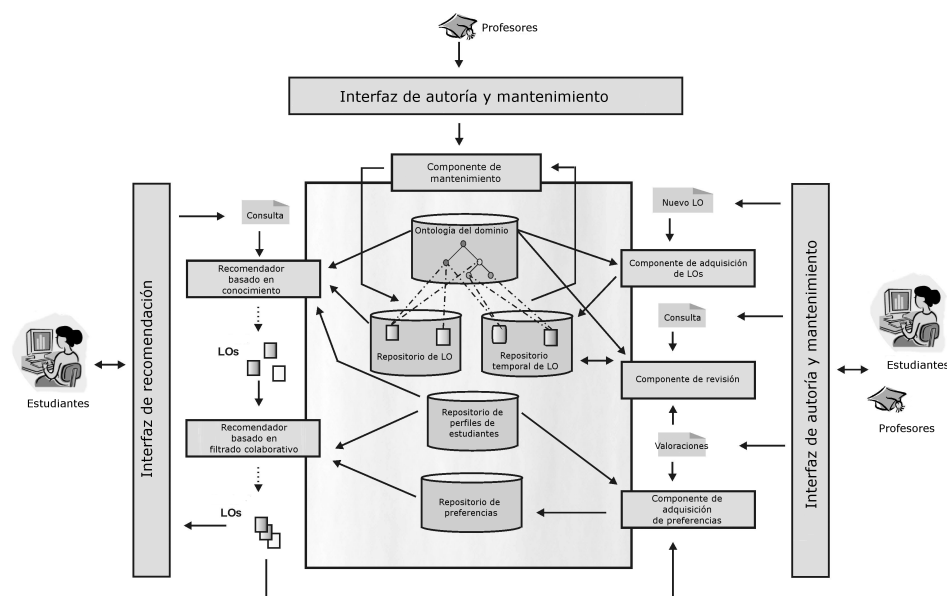


Figura 3.1: Esquema de la aproximación del modelo de gestión de repositorios de objetos de aprendizaje (se usan las siglas LO –del inglés *Learning Object*– para referirse a los recursos educativos).

3.3. Las fuentes de conocimiento

El uso de recomendadores en el ámbito educativo impone unos requisitos específicos y es posible sacar partido de diversos tipos de conocimiento en el proceso de recomendación (Drachslar et al., 2007, 2008). Por ejemplo, los recomendadores pueden sacar provecho del estado cognitivo del estudiante, el cual cambia a lo largo del tiempo. Esto permitiría incrementar el nivel de personalización a largo plazo. Por otro lado, los buenos itinerarios y estrategias de aprendizaje también puede proporcionar buenas guías para el recomendador. Por ejemplo, el recomendador puede sacar provecho de una regla pedagógica simple como “ir de tareas fáciles a difíciles” o “reducir de manera gradual la cantidad de orientación”. Los itinerarios de aprendizaje pueden representar rutas y secuencias diseñadas por los profesores a partir de experiencias positivas en el aula, o se pueden corresponder con el comportamiento de estudiantes avanzados.

En esta sección describimos de dónde proviene el conocimiento al que sacarán partido las estrategias de recomendación propuestas en este trabajo de tesis: la ontología del dominio (Sección 3.3.1), los objetos de aprendizaje y sus metadatos (Sección 3.3.2) y por último la información contextual (Sección 3.3.3).

3.3.1. La ontología del dominio

El término ontología en informática hace referencia a la formulación de un esquema conceptual y detallado del conocimiento asociado a un dominio concreto (Gruber, 1993). En el contexto del intercambio de conocimiento una ontología es un intento de formular un esquema conceptual dentro de un dominio dado, de manera que pueda ser reutilizable por cualquier otro sistema que trabaje sobre el mismo dominio.

La manera tradicional de formalizar este esquema consiste en identificar los conceptos fundamentales que componen el dominio y, a partir de ahí, establecer relaciones entre los mismos. Por lo tanto, una ontología no son sólo conceptos, también se compone de relaciones que representan un tipo de interacción entre los conceptos del dominio, de funciones que son consideradas como un tipo especial de relación y axiomas que son proposiciones siempre verdaderas y que se expresan en un lenguaje lógico. Con todos estos componentes las ontologías proporcionan un esquema que permite incluir conocimiento sobre la similitud entre los conceptos del dominio. Esta característica las hace idóneas para albergar el conocimiento de nuestras estrategias de recomendación, pudiendo modelar el dominio educativo correspondiente con los conceptos que lo fundamentan.

La ontología contiene una representación de los conceptos fundamentales que componen el dominio de aplicación. Por ejemplo, si el dominio son las matemáticas existirán conceptos como número real, suma, resta, etc. También puede permitir establecer cuáles son los objetivos pedagógicos que debe conseguir el estudiante. Esta representación se lleva a cabo mediante los conceptos –previamente definidos– que el estudiante debe ser capaz de dominar después del proceso de aprendizaje. En particular los objetivos pedagógicos serán un subconjunto de los conceptos del dominio. Por último, la ontología permite establecer una completa taxonomía de los conceptos del dominio si los conceptos son complejos, estructurando estos de manera jerárquica.

En las secciones posteriores veremos como, además, la ontología sirve de método de indexación de los objetos de aprendizaje y sirve para contener conocimiento sobre la información contextual utilizada por las estrategias de recomendación.

3.3.2. Objetos de aprendizaje

Como ya adelantamos en el capítulo sobre el estado del arte, un objeto de aprendizaje es un recurso educativo (generalmente digital) que puede ser usado y reutilizado como soporte a la enseñanza y al aprendizaje y que además es una experiencia de aprendizaje que contiene un objetivo y una actividad de aprendizaje. A lo largo de esta sección haremos un análisis del estándar de especificación escogido para representar los objetos de aprendizaje y de cómo se ha realizado la integración de los objetos de aprendizaje

en el conocimiento del recomendador.

Para el desarrollo de los objetos de aprendizaje se ha utilizado el estándar Learning Object Metadata (LOM) (IEEE, 15 July 2002), un modelo de datos, usualmente codificado en XML, usado para describir un objeto de aprendizaje y otros recursos digitales similares utilizados para el apoyo al aprendizaje. LOM es un estándar abierto, internacionalmente reconocido, que permite etiquetar con metadatos los objetos de aprendizaje de acuerdo a un conjunto predefinido de categorías. El principal propósito del estándar es ayudar a la reutilización de los objetos de aprendizaje. El modelo de datos de LOM especifica qué aspectos de un objeto de aprendizaje deberían ser descritos y qué vocabulario debe ser utilizado para esas descripciones; además define cómo este modelo de datos puede ser adaptado a las necesidades concretas con adiciones o restricciones.

LOM proporciona flexibilidad para utilizar el estándar, ya que no es necesario para la definición de los objetos de aprendizaje utilizar todos los elementos descritos en el modelo de datos y tampoco es necesario limitar la información que el estándar nos proporciona. Esto quiere decir que LOM permite a cada comunidad de usuarios utilizar su propio vocabulario para determinados elementos. Es por esto que en este trabajo se ha optado por adaptar el esquema LOM para el desarrollo de nuestros objetos de aprendizaje utilizando en el más alto nivel las siguientes categorías propuestas en el esquema: *general*, *ciclo de vida*, *técnica*, *educativa* y *relación*.

General Esta categoría contiene la información básica relativa al objeto de aprendizaje (título, descripción, idioma). Junto con esta información básica también contiene un elemento importante para la estrategia de recomendación: el metadato *keyword*. Dicho metadato contendrá la información relativa sobre los conceptos del dominio que son cubiertos por un objeto de aprendizaje. Un objeto de aprendizaje podrá contener tantas instancias de este metadato como sea necesario. Cada instancia del metadato *keyword* sirve como punto de enlace entre los objetos de aprendizaje contenidos en el repositorio y algunos de los conceptos del dominio que constituyen la ontología.

Ciclo de vida Esta categoría identifica al autor y el estado actual en el proceso de desarrollo del objeto de aprendizaje.

Técnica Esta categoría almacena la información relativa a la naturaleza del recurso (si es un documento de texto, imagen, etc.) así como la localización física del mismo.

Educativa La categoría *Educativa* nos permite identificar el tipo de recurso educativo al que pertenece el objeto de aprendizaje (por ejemplo ejercicios resueltos, ejercicios de desarrollo completo, ejercicios “encuentra el error”, cuestiones, etc.).

Relación La categoría *Relación* nos permite identificar si un objeto de aprendizaje es una versión mejorada o basada en otro objeto de aprendizaje contenido en el repositorio.

3.3.3. Información contextual

En este trabajo de tesis proponemos la inclusión de dos tipos de información contextual en el proceso de recomendación: contexto de la actividad y contexto del estudiante.

Contexto de la actividad: Esta información contextual está relacionada con la inclusión de itinerarios/caminos de aprendizaje que establecen un orden o precedencia en el que deben aprenderse los conceptos del dominio (por ejemplo, volviendo al dominio de las matemáticas, un itinerario de aprendizaje podría establecer la necesidad de saber sumar y restar antes de multiplicar). Los caminos de aprendizaje representan un papel muy importante en cualquier proceso de enseñanza. De ahí que otra buena fuente de información en un proceso de recomendación de recursos educativos sean dichos caminos de aprendizaje. Un itinerario de aprendizaje puede, por ejemplo, usarse para filtrar –descartar– o penalizar aquellos objetos de aprendizaje que contengan conceptos no alcanzables para el estudiante, es decir aquellos conceptos que de acuerdo al estado cognitivo del estudiante deben estudiarse más adelante. También puede definirse un camino de aprendizaje basado en hitos temporales y que pueda servir para filtrar o penalizar aquellos objetos de aprendizaje que cubran conceptos que no puedan estudiarse en una fecha dada. Esta información contextual es estática, debe ser definida por el instructor del curso antes de que los estudiantes utilicen el recomendador y puede incluir aquellas reglas pedagógicas que éste considere oportunas. La información contextual de la actividad estará representada en la ontología del dominio mediante relaciones de precedencia entre los conceptos que forman el camino de aprendizaje.

Contexto del estudiante: Esta información contextual está relacionada con los objetivos alcanzados por el estudiante durante su proceso de aprendizaje. Estos objetivos están representados por conceptos que el estudiante debe conocer junto con el nivel de competencia o conocimiento alcanzado en cada uno de ellos. Esta información contextual, al contrario de lo que sucedía con la anterior, evoluciona con el tiempo a medida que el estudiante interactúa con el recomendador y avanza en su aprendizaje. Se ha decidido usar un modelado *overlay* (o superpuesto) basado en ontología para representar el contexto del estudiante. De esta forma, este conocimiento se modela como una capa de pesos sobre la estructura de grano fino de la ontología: se representa por los conceptos de bajo nivel de la ontología y el nivel de competencia en cada

uno de ellos. Diversos investigadores reconocen los beneficios de este tipo de modelado (Sosnovsky y Dicheva, 2010).

La información contextual en el proceso de recomendación puede ser utilizada como un criterio más o menos estricto para seleccionar los objetos más relevantes. Un criterio *estricto* la utilizaría, por ejemplo, para limitar el conjunto de elementos disponibles: aquellos elementos que no cumplen los criterios son descartados. Un criterio *débil* la emplearía, por ejemplo, para ordenar el conjunto de los elementos ya seleccionados o para calcular una puntuación de relevancia para cada elemento. Como veremos en las siguientes secciones, en este trabajo de tesis se ha explorado el uso del contexto de la actividad como un criterio estricto con el fin de descartar aquellos objetos de aprendizaje que no son adecuados para el estudiante. Por otro lado, el uso del contexto del estudiante se ha empleado como un criterio débil que permitirá al recomendador asignar utilidades a cada objeto de aprendizaje.

3.4. Estrategia de recomendación basada en conocimiento sensible al contexto que promueve altos niveles de personalización

Esta primera estrategia de recomendación (Ruiz-Iniesta et al., 2009a,c, 2010) pretende extender y mejorar la estrategia basada en conocimiento que se describía en Gómez-Albarrán y Jiménez-Díaz (2009) superando la primera debilidad achacable que se ha mencionado en la Sección 3.1: la existencia de una personalización débil. Para ello se introduce una personalización fuerte, esto es, se incorpora la consecución de objetivos a largo plazo y la consecución de objetivos a corto plazo. La consecución de objetivos a corto plazo implica satisfacer, de forma más o menos aproximada, la consulta formulada por el estudiante. Por su parte la consecución de objetivos a largo plazo implica mejorar el nivel de competencia del alumno en los objetivos de aprendizaje establecidos por el instructor.

La estrategia basada en conocimiento que proponemos mantiene un comportamiento reactivo: el estudiante proporciona una consulta explícita al sistema y el recomendador responde con una recomendación. La consulta que realiza el estudiante está basada en los conceptos existentes en la ontología del dominio por lo que es importante que el estudiante conozca cuáles son estos conceptos para poder expresar sus necesidades en forma de consulta. Esta consulta representa los objetivos de la sesión de estudio para el estudiante: aquellos conceptos que quiere aprender en la sesión.

La respuesta de la recomendación se obtiene mediante tres procesos diferenciados: recuperación, filtrado y valoración.

El proceso de recuperación comienza buscando en el repositorio un conjunto inicial de objetos de aprendizaje que satisfagan de una manera apro-

ximada la consulta del estudiante. Este proceso primero intenta encontrar objetos de aprendizaje indexados exactamente por los conceptos de la consulta. Si no hay objetos de aprendizaje que satisfagan esta condición, o si estamos interesados en una localización más flexible, los objetos de aprendizaje indexados por un subconjunto de los conceptos (iguales o similares a los) especificados en la consulta del estudiante serán recuperados. Para ello se utilizará la taxonomía de conceptos incluida en la ontología. La recuperación aproximada contempla la posibilidad de recuperar objetos de aprendizaje que cubran conceptos hermanos de aquellos que están representados en la consulta o con un grado menor de parentesco.

Seguidamente, en el proceso de filtrado se descartan del conjunto de objetos de aprendizaje recuperados aquellos recursos que no sean apropiados para el estudiante actual. El objetivo de este proceso de filtrado es descartar aquellos objetos de aprendizaje indexados por conceptos de la ontología no alcanzables para el estudiante actual y conseguir así aumentar el nivel de personalización. Para ello, la estrategia hace uso de la información contextual sobre los conceptos del dominio –contexto de la actividad– y el estado cognitivo del estudiante –contexto del estudiante.

La combinación de los dos tipos de información contextual nos permite saber qué objetos de aprendizaje de los inicialmente recuperados cubren conceptos no alcanzables por el estudiante actual, con el fin de dejar en el conjunto sólo aquellos objetos que cubran conceptos “listos para ser explorados” por dicho estudiante. En este trabajo de tesis se ha considerado que un concepto está “listo para ser explorado” por el estudiante si, de acuerdo a su contexto de estudiante –el nivel de conocimiento alcanzado en los objetivos de aprendizaje– y al contexto de la actividad –el camino de aprendizaje definido sobre los conceptos la ontología– cumple alguna de las siguientes condiciones:

- Es un concepto ya explorado por el estudiante, por lo tanto aparecerá en su contexto de estudiante con su correspondiente nivel de conocimiento.
- Es un concepto que el estudiante todavía no ha explorado pero puede ser uno de los siguientes a aprender. Si un concepto c_1 precede a un concepto c_2 en el itinerario de aprendizaje, un estudiante puede descubrir c_2 si el nivel de competencia del estudiante alcanzado en c_1 supera un cierto *umbral de progreso*. Si varios conceptos c_1, c_2, \dots, c_k son predecesores directos de un concepto c_x , consideramos que c_x puede ser descubierto si la media aritmética de los niveles de competencia de todos los predecesores directos superan (o igualan) el umbral de progreso.

Este proceso de filtrado proporciona una forma de personalización a largo plazo. De esta manera, ante una misma consulta formulada por dos es-

tudiantes distintos, el conjunto de objetos de aprendizaje candidatos tras el filtrado podría variar significativamente en función del dominio de la materia que tenga cada uno.

Finalmente, el proceso de valoración asigna una relevancia, que llamaremos *calidad*, a cada objeto de aprendizaje obtenido en el proceso anterior y los ordena de acuerdo a dicho valor.

A fin de calcular la calidad de un objeto de aprendizaje L para un estudiante S que ha proporcionado una consulta Q ($Calidad(L, S, Q)$) hemos decidido ensayar distintas métricas de calidad que combinan dos relevancias parciales: la similitud entre Q y los conceptos cubiertos por L ($Similitud(L, Q)$), y la utilidad pedagógica de L con respecto al estudiante S ($UP(L, S)$). Las métricas escogidas siguen patrones básicos de agregación de preferencias usados a menudo en los sistemas de recomendación (Beliakov et al., 2011). El recomendador sólo hará uso de una de ellas, sin embargo, hemos decidido emplear dos métricas diferentes a fin de estudiar cuál de ellas obtiene mejores niveles de calidad en la lista final de objetos de aprendizaje. Las métricas empleadas son las siguientes:

$$Calidad(L, S, Q) = \alpha \cdot Similitud(L, Q) + (1 - \alpha) \cdot UP(L, S) \quad \text{donde } \alpha \in [0, 1] \quad (3.1)$$

$$Calidad(L, S, Q) = \frac{1}{\frac{\alpha}{Similitud(L, Q)} + \frac{(1-\alpha)}{UP(L, S)}} \quad \text{donde } \alpha \in [0, 1] \quad (3.2)$$

El cómputo de la $Similitud(L, Q)$ entre los conceptos recogidos en la consulta Q y los conceptos que L cubre requiere calcular la similitud entre dos conjuntos de conceptos. Existen diferentes métricas en la literatura que pueden usarse para calcular la similitud entre conjuntos de valores. En este trabajo hemos decidido realizar una simplificación que consiste en comparar los conceptos individuales que resultan de la conjunción de cada uno de los dos conjuntos de conceptos –el concepto que resulta de la conjunción de los conceptos de la consulta (Q_conj_c) y el concepto que resulta de la conjunción de los conceptos que cubre L (L_conj_c)– en lugar de comparar directamente dos conjuntos de conceptos. Asimismo, teniendo en cuenta que los conceptos están organizados jerárquicamente, hemos decidido utilizar la métrica de similitud definida en (González-Calero et al., 1999) que permite sacar partido a la clasificación taxonómica de los conceptos:

$$Similitud(L, Q) = \frac{|super(Q_conj_c) \cap super(L_conj_c)|}{\sqrt{|super(Q_conj_c)|} \cdot \sqrt{|super(L_conj_c)|}} \quad (3.3)$$

donde $super(Q_conj_c)$ representa el conjunto de todos los conceptos contenidos en la ontología que son superconceptos de Q_conj_c y $super(L_conj_c)$ contiene todos los conceptos contenidos en la ontología que son superconceptos de L_conj_c .

El valor de $Similitud(L, Q)$ se encuentra en el intervalo $[0, 1]$. La similitud calcula la relevancia de un objeto de aprendizaje debida a los objetivos de la sesión (reflejados en la consulta propuesta por el estudiante) que L satisface, lo que hemos llamado los objetivos a corto plazo. Cuanto mayor es el número de conceptos de la consulta que L permite aprender, mayor será el valor de la similitud. Cuanto más similares sean los conceptos que cubre L y los conceptos explicitados en la consulta, mayor será el valor de similitud. Como podemos observar, el conocimiento de la similitud entre los conceptos representados en la ontología es crucial en nuestro contexto.

La segunda relevancia considerada, la utilidad pedagógica, se refiere a lo adecuado que es el objeto de aprendizaje para el estudiante en función de la información contextual relativa al mismo. A fin de medir la utilidad pedagógica que el objeto de aprendizaje L muestra para un estudiante S dado, $UP(L, S)$, hemos adoptado una estrategia instructiva que promueve completar las carencias de conocimiento del estudiante incluyendo conocimientos de refuerzo (Siemer y Angelides, 1998). El objetivo es asignar una utilidad pedagógica mayor a L si éste cubre conceptos en los cuales el estudiante ha demostrado tener un bajo nivel de competencia. De esta manera, L puede ayudar al estudiante a reforzar sus conocimientos sobre esos conceptos y así alcanzar sus objetivos de aprendizaje a largo plazo. Este refuerzo de conocimiento podría asignar prioridad a conceptos que el estudiante todavía no ha explorado, o puede repartirla de una manera igualitaria entre los conceptos explorados y no explorados. Hemos optado por calcular la utilidad pedagógica de acuerdo a la siguiente fórmula.

$$UP(L, S) = 1 - MAN(L, S) \quad (3.4)$$

donde $MAN(L, S)$ es la media aritmética normalizada de los niveles de competencia alcanzados por el estudiante S en los conceptos cubiertos por L , de modo que $MAN(L, S)$ estará en el intervalo $[0, 1]$. En consecuencia $UP(L, S)$ también toma valores entre 0 y 1.

Finalmente este conjunto de objetos de aprendizaje será ordenado decrecientemente por la calidad $Calidad(L, S, Q)$ asignada a cada uno de ellos y aquellos que ocupen las k primeras posiciones en la lista ordenada serán seleccionados y recomendados al estudiante.

Si nos fijamos en las dos métricas de calidad presentadas en esta sección, Ecuaciones (3.1) y (3.2), valores bajos de α dan mayor peso a la utilidad pedagógica en contra de la similitud con la consulta. En particular, $\alpha = 0$ representa el nivel más alto de personalización a largo plazo, y en este caso, la consulta (objetivos para la sesión) sólo se utiliza en el proceso de recu-

peración. Esto asegura que el recomendador propone objetos de aprendizaje que cumplan con los objetivos de la sesión a un nivel mínimo, aunque el orden en el cual son propuestos al estudiante está totalmente influenciado por los objetivos a largo plazo que permiten alcanzar (su capacidad de refuerzo de conocimientos). Por el contrario, valores altos de α dan la prioridad a la similitud con la consulta frente a la utilidad pedagógica.

La introducción de la información contextual relativa al estudiante en el cálculo de la utilidad pedagógica y, como consecuencia, en el cálculo de la calidad, permite la incorporación de un notable nivel de personalización en la estrategia de recomendación. Como ya hemos indicado, la influencia final del contexto del estudiante depende del parámetro α en las Ecuaciones (3.1) y (3.2). Más adelante, en la Sección 5.2, se realizará una evaluación de la influencia que tienen el parámetro α y la métrica de calidad seleccionada en los resultados de la recomendación junto con un análisis sobre el tamaño final del conjunto recomendado (parámetro k).

3.5. Estrategia de recomendación basada en conocimiento que promueve la diversidad

Esta segunda estrategia (Ruiz-Iniesta et al., 2011a) surge con la intención de solventar el problema de la sobreespecialización que aparece en la primera aproximación (Gómez-Albarrán y Jiménez-Díaz, 2009). Solventar el problema de la sobreespecialización e introducir diversidad en las recomendaciones es crucial para poder aprovechar al máximo cada recomendación. Como ya vimos en el capítulo dedicado al estado del arte, podemos encontrar trabajos que defienden esta idea y proponen diferentes algoritmos genéricos para introducir diversidad en las recomendaciones. Para conseguir introducir diversidad en la recomendación propuesta hemos adaptado el algoritmo de diversidad genérico propuesto por (Smyth y McClave, 2001) a nuestro dominio concreto de aplicación. Esta segunda estrategia basada en conocimiento que proponemos mantiene un comportamiento reactivo, al igual que la explicada en el punto anterior: el estudiante proporciona una consulta explícita al sistema (usando los conceptos existentes en la ontología del dominio y representando los objetivos de la sesión de estudio para el estudiante) y el recomendador responde con una recomendación.

En esta estrategia, la respuesta de la recomendación se obtiene mediante cuatro procesos: recuperación, filtrado, valoración y selección. Los procesos de recuperación y filtrado son similares a los presentados en la Sección 3.4 y por tanto permite satisfacer la consulta formulada e introduce cierta personalización gracias al proceso de filtrado. Por su parte la etapa de valoración le asigna a cada objeto de aprendizaje una puntuación basada en la similitud que éste muestra con la consulta usando la métrica de similitud descrita en la Ecuación 3.3. Finalmente, el proceso de selección es el encargado de introdu-

cir diversidad en el conjunto final de objetos de aprendizaje recomendados.

Esta estrategia propone un proceso de selección basado en diversidad que hace uso de una métrica de calidad que promueve la diversidad en el conjunto de objetos de aprendizaje recomendados. Esta métrica de calidad asigna la prioridad a aquellos objetos de aprendizaje que son más similares a la consulta planteada por el estudiante y, al mismo tiempo, son disimilares con el resto de objetos de aprendizaje ya seleccionados para ser recomendados.

Este proceso de selección, que hemos denominado selección basada en diversidad, adapta el algoritmo *Bounded Greedy Selection* propuesto por (Smyth y McClave, 2001). El algoritmo de diversidad es el mostrado a continuación, donde Q representa la consulta, R es el conjunto de objetos de aprendizaje candidatos a ser recomendados y k es el tamaño del conjunto final de objetos de aprendizaje que finalmente serán recomendados al estudiante (P):

Algoritmo 1 Algoritmo de selección basado en diversidad

Entrada: Q, R, k

Salida: P

```

1:  $P := \{\}$ 
2: para  $i := 1$  hasta  $k$  hacer
3:   Ordenar  $R$  por  $\text{Calidad}(L, P, Q)$  para cada  $L$  en  $R$ 
4:    $P := P + \text{primero}(R)$ 
5:    $R := R - \text{primero}(R)$ 
6: fin para
7: devolver  $P$ 

```

En resumen, la selección basada en diversidad parte de un conjunto R , que en nuestro caso es el resultado de los procesos de recuperación y filtrado, y de manera incremental construye un conjunto P de k objetos de aprendizaje que finalmente serán los propuestos al estudiante. Durante cada paso, el resto de objetos de aprendizaje que se encuentran en R son ordenados por una métrica de calidad, $\text{Calidad}(L, P, Q)$ que intentará primar la diversidad, y aquel objeto de aprendizaje con mayor calidad es añadido a P y eliminado de R .

Para el cálculo de la calidad de un objeto de aprendizaje L , hemos decidido probar dos métricas de calidad que combinan dos relevancias –la similitud entre Q y los conceptos que L cubre, $\text{Similitud}(L, Q)$, y la diversidad de L relativa a los objetos de aprendizaje ya seleccionados para ser recomendados $\text{RelD}(L, P)$ – siguiendo los mismos patrones de agregación que usamos en las Ecuaciones (3.1) y (3.2). Estas métricas quedan como sigue:

$$\text{Calidad}(L, P, Q) = \alpha \cdot \text{Similitud}(L, Q) + (1 - \alpha) \cdot \text{RelD}(L, P)$$

donde $\alpha \in [0, 1]$ (3.5)

$$Calidad(L, P, Q) = \frac{1}{\frac{\alpha}{Similitud(L, Q)} + \frac{(1-\alpha)}{RelD(L, P)}} \quad \text{donde } \alpha \in [0, 1] \quad (3.6)$$

Para el cálculo de la diversidad relativa $RelD$, hemos adoptado la métrica propuesta en (Smyth y McClave, 2001):

$$RelD(L, P) = \begin{cases} 1 & \text{si } P = \{\} \\ \frac{\sum_{L_i \in P} (1 - Similitud(L, L_i))}{|P|} & \text{en otro caso} \end{cases} \quad (3.7)$$

siendo $Similitud(L, L_i)$ la similitud entre los conjuntos de conceptos cubiertos por L y L_i . De nuevo, en la Sección 5.2 mostraremos el comportamiento de esta estrategia de recomendación haciendo uso de ambas métricas de calidad.

3.6. Estrategia de recomendación basada en conocimiento que combina proactividad y navegación por propuesta

En esta última estrategia (Ruiz-Iniesta et al., 2009b) se ha aprovechado para ensayar una aproximación proactiva de las recomendaciones, como alternativa a la naturaleza reactiva de las otras estrategias diseñadas, que se complementa con una navegación por propuesta que permite recibir realimentación del usuario sin que suponga mucho esfuerzo para él. El objetivo de esta estrategia es proporcionar una ayuda más adecuada a aquellos estudiantes que tienen una actitud pasiva o que tienen poco conocimiento del dominio como para proponer una consulta al recomendador.

En esta estrategia el estudiante, al comenzar una sesión de recomendación, recibirá un conjunto de objetos de aprendizaje propuestos como actividades del día, es decir, un conjunto de recursos educativos que se proponen al estudiante para que continúe con su proceso de aprendizaje. Ante esta primera recomendación el estudiante tiene dos opciones: seleccionar un objeto de aprendizaje para trabajar en él, o mostrar sus preferencias (seleccionando un objeto de aprendizaje y diciendo que quiere “más como éste”) para refinar la propuesta. Si se da la primera opción llegamos al final del proceso de recomendación. Si, por el contrario, el alumno proporciona una preferencia para refinar la recomendación, el recomendador propondrá de nuevo otro conjunto de objetos de aprendizaje al estudiante como resultado del refinamiento del conjunto anterior.

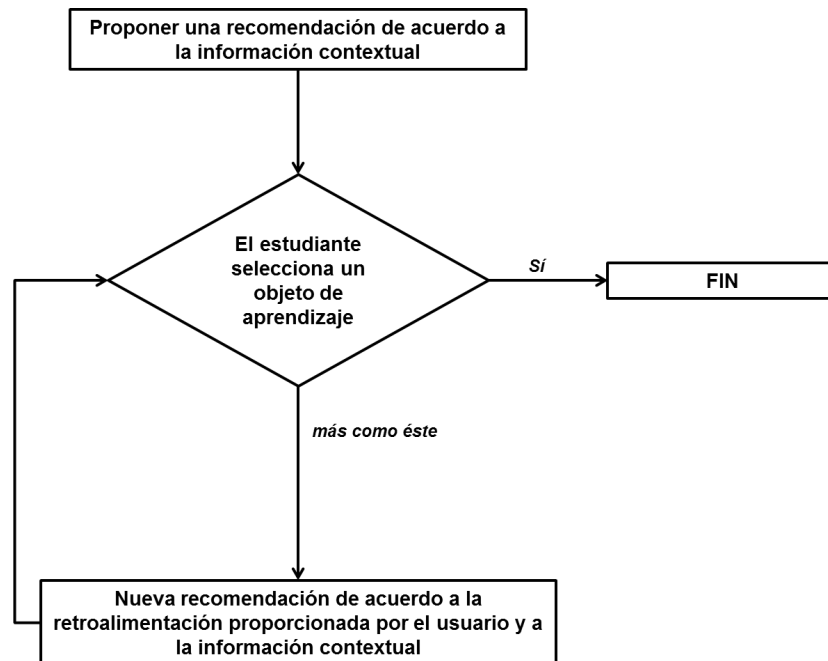


Figura 3.2: Diagrama de interacción usuario-recomendador para la estrategia proactiva.

En la Figura 3.2 podemos ver un diagrama de funcionamiento de esta estrategia, que podemos dividir en dos fases: la realización de la primera propuesta, y la realización de la segunda y sucesivas propuestas. Como veremos, las propuestas resultan de realizar un recorrido por niveles en la taxonomía de conceptos de la ontología, y de tener en cuenta la información contextual (y la realimentación del usuario, en caso de existir).

Presentamos a continuación el funcionamiento de cada fase.

- En la primera fase, el repositorio se divide, de acuerdo a la información contextual, en dos grupos de objetos de aprendizaje: objetos de aprendizaje que cubren conceptos ya estudiados, que los identificaremos como *objetos de aprendizaje de refuerzo*, y objetos de aprendizaje que cubren nuevos conceptos que pueden ser explorados por el estudiante, identificados como *objetos de aprendizaje de descubrimiento*. Para recuperar cada grupo de objetos de aprendizaje es necesario primero identificar cuáles son los conceptos ya estudiados y cuáles los nuevos, para lo que se usa la información contextual de la actividad y la información contextual del estudiante. Una vez identificados los conceptos, se genera una consulta formada por el concepto de más alto nivel en la

jerarquía que subsume a todos los conceptos de *refuerzo* y otra consulta formada por el concepto de más alto nivel de la jerarquía que subsume a los conceptos de *descubrimiento*. A continuación la estrategia recupera dos grupos (uno para cada consulta) de objetos de aprendizaje (los de refuerzo y los de descubrimiento). Para adaptar los resultados al estudiante actual y conseguir así un mayor nivel de personalización, cada uno de los conjuntos de objetos de aprendizaje recuperados se filtra y se descartan aquellos objetos de aprendizaje que no están listos para ser explorados (ver Sección 3.4). A continuación ambos conjuntos son ordenados de acuerdo a una métrica de calidad y se selecciona al menos un representante de cada conjunto. Estos representantes son los que finalmente serán mostrados al estudiante.

- En la segunda fase, a la cual llegamos si el estudiante ha decidido seguir refinando la propuesta de la primera fase, se produce un recorrido por niveles en la taxonomía de conceptos del dominio. Cada nueva propuesta estará constituida por objetos de aprendizaje que cubran conceptos del nivel más alto de la taxonomía dentro del grupo al que pertenecía el objeto de aprendizaje elegido para refinar. En la recuperación el repositorio se divide en tantos grupos como conceptos existan en el nivel de la taxonomía en el que nos encontremos. Dicho de otra manera, para realizar una propuesta, en esta fase se restringe la recuperación a la rama de la taxonomía de conceptos encabezada por el concepto C del que era representante el objeto de aprendizaje elegido por el alumno al seleccionar “más como éste”. Para recuperar estos grupos se generan tantas consultas como subconceptos tenga el concepto C y se realiza una recuperación por consulta. Cada uno de estos grupos es filtrado para descartar objetos de aprendizaje que cubren conceptos que no están listos para ser explorados para el estudiante actual. A continuación cada uno de estos conjuntos es ordenado de acuerdo a una métrica de calidad y se selecciona al menos un representante de cada conjunto para después ser propuestos al estudiante. Este mecanismo sirve tanto para la segunda propuesta como para las sucesivas, lo que permite ir descendiendo por la taxonomía e ir estrechando la búsqueda a zonas concretas de la misma.

Una vez explicado el funcionamiento de la estrategia sólo nos queda decidir la métrica que asignará calidad a cada objeto de aprendizaje L . En esta estrategia hemos decidido explorar el uso de dos métricas de calidad que combinan dos relevancias parciales: la similitud entre la consulta Q autoformulada por la estrategia y los conceptos cubiertos por L ($Similitud(L, Q)$), y la utilidad pedagógica de L con respecto al estudiante S ($UP(L, S)$):

$$Calidad(L, S, Q) = \alpha \cdot Similitud(L, Q) + (1 - \alpha) \cdot UP(L, S) \quad \text{donde } \alpha \in [0, 1] \quad (3.8)$$

$$Calidad(L, S, Q) = \frac{1}{\frac{\alpha}{Similitud(L, Q)} + \frac{(1-\alpha)}{UP(L, S)}} \quad \text{donde } \alpha \in [0, 1] \quad (3.9)$$

Aunque en esta tercera estrategia hemos optado por explorar métricas de calidad que combinen la similitud con la consulta y la utilidad pedagógica, es interesante señalar que el proceso de recomendación podría emplear métricas de calidad que primasen la inclusión de diversidad en el conjunto recomendado.

Esta última estrategia de recomendación da respuesta a la última deficiencia de la estrategia inicial basada en conocimiento. Con este modelo de funcionamiento el recomendador engancha al estudiante en un proceso conversacional guiado por las preferencias que el propio estudiante va mostrando con respecto a cada recomendación. El motivo por el que hemos elegido un modelo de navegación por propuesta, en lugar de un modelo de navegación por preguntas, es por considerarla más apropiada en dominios complejos donde el usuario puede no estar suficientemente preparado para contestar preguntas sobre sus requisitos, ya que el conocimiento del dominio por parte del usuario puede ser insuficiente.

3.7. Resumen y conclusiones

En este capítulo hemos descrito tres estrategias de recomendación de objetos de aprendizaje que intentan afrontar los aspectos mejorables de la recomendación basada en conocimiento empleada en el enfoque híbrido descrito en (Gómez-Albarrán y Jiménez-Díaz, 2009). Antes de presentar cada estrategia se han descrito las fuentes de conocimiento necesarias: la ontología del dominio, los objetos de aprendizaje y la información contextual. Gracias a la inclusión de las fuentes de conocimiento y, particularmente, la inclusión y el uso de la información contextual en distintos procesos de las estrategias de recomendación, estas estrategias producen recomendaciones más personalizadas. Estas tres estrategias basadas en conocimiento constituyen un enfoque muy novedoso en el acceso personalizado a recursos educativos contenidos en repositorios y dan respuesta a las necesidades manifestadas por los usuarios de los repositorios de objetos de aprendizaje (Dichev y Dicheva, 2012). Al mismo tiempo estas tres estrategias pueden sustituir a la

correspondiente basada en conocimiento existente en el recomendador híbrido descrito por Gómez-Albarrán y Jiménez-Díaz (2009). Cualquiera de estas estrategias podría encajar dentro de ese recomendador híbrido para, a través de una segunda estrategia de filtrado colaborativo, refinar el conjunto propuesto mediante las valoraciones otorgadas por los estudiantes a los distintos objetos de aprendizaje

Las estrategias sacan partido a una descripción de los recursos educativos que aumenta el potencial de estándares de metadatos como LOM con una indexación semántica basada en la ontología. En este sentido coincidimos con otros autores que usan descripciones semánticas para facilitar el descubrimiento de recursos educativos digitales (Piedra et al., 2010).

Como hemos indicado, la primera de las estrategias presentada en la Sección 3.4 (Ruiz-Iniesta et al., 2009a,c, 2010) permite incorporar personalización en los procesos que producen la recomendación: en el proceso de recuperación, teniendo que satisfacer los objetivos a corto plazo recogidos en la consulta; superando un proceso de filtrado que excluye objetos de aprendizaje no aptos para el estudiante; y en la valoración, donde se emplea una métrica de calidad que combina similitud con la consulta y utilidad pedagógica de los objetos de aprendizaje recuperados. La inclusión de la utilidad pedagógica permite reforzar aún más la consecución de objetivos a largo plazo, y, en definitiva, la personalización o adaptación al estudiante. Para valorar la utilidad pedagógica se ha seguido una estrategia instructiva que, como forma de conseguir objetivos a largo plazo, promueve suplir las deficiencias de conocimiento del estudiante, valorando mejor a aquellos objetos de aprendizaje recuperados que permitan reforzar el conocimiento de los conceptos en los que el estudiante ha demostrado un bajo nivel de competencia. Evidentemente, esta estrategia de refuerzo, como forma de incluir objetivos a largo plazo, sería intercambiable en el marco definido por otras estrategias instructivas sin más que cambiar la forma de valorar la utilidad pedagógica.

La segunda estrategia presentada promueve la diversidad entre los objetos de aprendizaje recomendados a la vez que preserva los objetivos a corto plazo del estudiante y sigue realizando un proceso de filtrado que tiene en cuenta los objetivos a largo plazo (Ruiz-Iniesta et al., 2011a). La respuesta de la recomendación se obtiene mediante cuatro procesos: recuperación, filtrado, valoración y selección. Los procesos de recuperación y filtrado son similares a los presentados en la primera estrategia. Por su parte la etapa de valoración asigna a cada objeto de aprendizaje una puntuación basada en la similitud que éste muestra con la consulta. Finalmente, el proceso de selección es el encargado de introducir diversidad en el conjunto final de objetos de aprendizaje recomendados. La influencia final de la diversidad en el conjunto final puede ser controlado en las métricas de calidad empleadas —que combinan la similitud con la consulta computada en la etapa de valoración y la diversidad relativa entre objetos de aprendizaje— para asignar relevancia

a cada objeto de aprendizaje.

Por su parte, la estrategia descrita en la Sección 3.6 (Ruiz-Iniesta et al., 2009a) combina proactividad y navegación por propuesta como modelo de interacción con el estudiante, afrontando así la última potencial deficiencia de la estrategia inicial basada en conocimiento: la naturaleza reactiva de la estrategia, que impone que el estudiante tenga que realizar una consulta. Esta navegación engancha al estudiante en un proceso conversacional en el que podrá refinar el conjunto propuesto. Se espera que el modelo de interacción propuesto ayude a estudiantes poco motivados, o que no tienen un conocimiento suficiente del dominio como para proponer una consulta, a que continúen su proceso de aprendizaje. Además gracias al uso de la información contextual –de estudiante y de actividad– y de las preferencias del estudiante recogidas en cada interacción se consigue personalización en las recomendaciones.

Una vez presentadas las estrategias de recomendación propuestas es necesario realizar una evaluación tanto computacional, que nos indique sus bondades, como una evaluación con profesores y estudiantes que ratifiquen la satisfacción con las mismas. Estas evaluaciones se presentarán en el capítulo 5. Para poder realizar estas evaluaciones es necesario realizar implementaciones de las estrategias. El desarrollo de prototipos para cada estrategia y la subsiguiente detección de aspectos comunes y ejes de variabilidad ha dado lugar a un framework orientado a objetos que facilita el prototipado rápido de sistemas de recomendación basados en conocimiento en el ámbito educativo. El próximo capítulo lo dedicamos a la descripción de este framework.

Capítulo 4

Un framework para el desarrollo de recomendadores basados en conocimiento en el ámbito educativo

*Esta máquina puede hacer cualquier cosa
que sepamos cómo ordenarle que la
ejecute.*

Ada Lovelace

Como mostramos en el capítulo anterior, en el trabajo en recomendación de objetos de aprendizaje que hemos tomado como punto de partida (Gómez-Albarrán y Jiménez-Díaz, 2009) encontramos algunas carencias, como la limitada personalización incluida en el proceso de recomendación, la ausencia de introducción de diversidad entre los objetos de aprendizaje finalmente recomendados y las dificultades que podía suponer para algunos estudiantes el hecho de tener que realizar una consulta al sistema para obtener una recomendación. Nuestros esfuerzos se centraron a partir de entonces en desarrollar estrategias que pudieran abordar las carencias detectadas. El resultado ha sido tres estrategias de recomendación que coinciden en varios aspectos. Todas ellas pertenecen al grupo de sistemas de recomendación basados en conocimiento y confían en la existencia de una base de conocimiento sobre las fuentes que intervienen en el proceso de recomendación, a saber, los objetos de aprendizaje, la información contextual y la ontología. Una vez obtenidas las estrategias, el siguiente paso era evaluar el comportamiento de las mismas analizando los recursos que recomienda cada una. Para llevar a cabo esta evaluación comenzamos el proceso de desarrollo de una serie de prototipos que pusieran en práctica las estrategias diseñadas. En este pro-

ceso detectamos que existían unos factores comunes de diseño en todas las estrategias. Estos factores nos han hecho posible abstraer un proceso de recomendación compuesto por cinco etapas y que tiene unos ejes de variabilidad que le otorgan una alta flexibilidad.

En un primer momento se ha realizado un estudio sobre qué frameworks y/o librerías existen que pudieran ayudarnos en el desarrollo de los prototipos necesarios. Este estudio, como veremos más adelante, concluyó en que no existía ningún framework y/o librería que se adaptara a un escenario de recomendación como el nuestro. Este hecho, unido al esfuerzo ya realizado de abstracción del proceso común de recomendación arriba mencionado, ha desembocado en la creación de nuestro propio framework destinado a facilitar el prototipado rápido de sistemas de recomendación basados en conocimiento en el ámbito educativo que hacen uso del conocimiento antes citado (Ruiz-Iniesta et al., 2011c, 2012a).

En el presente capítulo se describe cómo es el framework desarrollado, sus características y ejemplifica la implementación de recomendadores concretos usando dicho framework. El capítulo está organizado de la siguiente manera. La Sección 4.1 presenta algunos frameworks existentes para la construcción de recomendadores haciendo un análisis de por qué estos frameworks no son de utilidad en escenarios de recomendación como los que se proponen en esta tesis. A continuación la Sección 4.2 describe las etapas genéricas de dichos escenarios de recomendación así como los distintos ejes de variabilidad. La Sección 4.3 describe el framework, detallando las clases que lo conforman (Sección 4.3.1), los puntos flexibles (Sección 4.3.2) y la implementación concreta de algunas de las clases abstractas (Sección 4.3.3). La Sección 4.4 presenta tres casos prácticos de utilización del framework. Terminaremos el capítulo con las conclusiones obtenidas (Sección 4.5).

4.1. Frameworks y librerías para la construcción de recomendadores

Se entiende por framework una arquitectura semi-completa reutilizable para el desarrollo de aplicaciones de un determinado dominio, que no sólo permite la reutilización de código sino también de diseño (Pree, 1994). En el ámbito de los recomendadores y desde el punto de vista tecnológico existen librerías y frameworks que facilitan el desarrollo de sistemas de recomendación, evitando tener que implementar ciertos algoritmos desde cero. Algunas de estas librerías son: SUGGEST¹, COFI², Apache Mahout³, Duine, MyMe-

¹<http://glaros.dtc.umn.edu/gkhome/suggest/overview>

²<http://www.nongnu.org/cofi/>

³<http://mahout.apache.org>

diaLite⁴, Crab⁵, GraphLab⁶ y Lenskit⁷. La mayoría de ellas proporcionan algoritmos de filtrado colaborativo en distintos lenguajes como PHP, Java o C#, entre otros.

SUGGEST y COFI fueron de las primeras librerías que se crearon para el desarrollo de sistemas de recomendación, pero a día de hoy ya no son mantenidas. Ambas se centraban en los algoritmos de filtrado colaborativo básicos: filtrado basado en usuario y basado en producto.

Apache Mahout es una librería desarrollada en Java que incluye una colección de los algoritmos de aprendizaje máquina y minería de datos más conocidos (Owen et al., 2011). Una parte de la librería está dedicada a los algoritmos de filtrado colaborativo, concretamente contiene algoritmos que siguen la aproximación de los k vecinos más cercanos y algoritmos basados en predictores Slope One (Lemire y Maclachlan, 2005; Lee et al., 2012).

Duine, como la anterior, es una librería desarrollada en Java centrada en desarrollar mecanismos de predicción para los algoritmos de recomendación. Un mecanismo de predicción es un componente que predice cómo de interesante es para un usuario una pieza de información (un elemento/producto). Duine produce predicciones en el rango -1 (nada interesante) hasta +1 (muy interesante). Además también procesa y almacena las valoraciones que un usuario ha asignado a un elemento y los intereses del usuario en cuanto a las características de los elementos que se está recomendando. Para calcular las predicciones Duine utiliza la información del usuario y la información de los elementos. Ofrece 7 algoritmos distintos para realizar el cálculo de las predicciones: filtrado colaborativo (o filtrado social), recuperación de información (extrae información de un elemento, hace un análisis del texto y empareja el resultado con el perfil del usuario para determinar lo interesante qué es el tema), razonamiento basado en casos (basado en el número de elementos similares que han sido recomendados al usuario en el pasado), GenrLMS (razonamiento sobre el género), TopNDeviation (basado en la popularidad de un elemento), AlreadyKnow y UserAverage. Duine también proporciona mecanismos para la evaluación de los recomendadores desarrollados, puede ser extendido y cuenta con una interfaz de programación de aplicaciones que proporciona explicaciones sobre cómo ha sido obtenida una predicción.

MyMediaLite (Gantner et al., 2011) es una librería desarrollada en C# que aborda dos de las tareas comunes: predicción de valoraciones (por ejemplo, en una escala de 1 a 5) y predicción de elementos con feedback positivo (por ejemplo, a partir de *clicks* o acciones de compra). En cuanto a los algoritmos implementados para llevar a cabo estas dos tareas están: los k vecinos más cercanos, métodos básicos como predictores Slope-One (Lemire

⁴<http://mymedialite.net/>

⁵<http://muricoca.github.com/crab/index.html>

⁶<http://graphlab.org>

⁷<http://lenskit.grouplens.org>

y Maclachlan, 2005) y medias y métodos más avanzados de factorización matricial (Koren et al., 2009).

Crab nació en abril de 2011 y permite el desarrollo de sistemas de recomendación en Python. Está concebida como una extensión de los paquetes científicos de Python. Proporciona algoritmos de filtrado colaborativo básicos: filtrado basado en usuario y basado en producto. Es un nuevo proyecto que está en marcha y dentro de su desarrollo está la inclusión de otros algoritmos de filtrado colaborativo (Lee et al., 2012).

GraphLab (Low et al., 2010) es una librería para la generación de sistemas de recomendación basados en filtrado colaborativo. Incluye múltiples algoritmos para mejorar la factorización de matrices. Algunos de estos algoritmos son: PMF (del inglés, *Probabilistic Matrix Factorization*) (Salakhutdinov y Mnih, 2008), BPTF (del inglés, *Bayesian Probabilistic Tensor Factorization*), ALS (del inglés, *Alternating Least Squares*), WALS (del inglés, *Weighted Alternating Least Squares*), SGD (del inglés, *Stochastic Gradient Descent*), SVD (del inglés, *Singular Value Decomposition*), NMF (del inglés, *Non-negative Matrix Factorization*) y el algoritmo de Koren SVD++ (Koren et al., 2009).

Una de las últimas en desarrollarse ha sido Lenskit pero ha cobrado fuerza en el panorama actual. Lenskit es un framework en Java para el desarrollo de recomendadores basados en filtrado colaborativo (Ekstrand et al., 2011). Este framework ha sido desarrollado dentro del grupo de investigación GroupLens, uno de los más activos dentro de los sistemas de recomendación, y ha sido implantado en MovieLens, aspectos que le proporcionan un gran respaldo tanto dentro de la comunidad investigadora como en la industria. Lenskit proporciona una API común para los sistemas de recomendación basados en filtrado colaborativo, implementaciones de varios algoritmos de filtrado colaborativo y un framework de evaluación para facilitar el desarrollo de evaluaciones off-line de los sistemas desarrollados.

Como hemos visto casi todas las librerías propuestas proporcionan métodos para el filtrado colaborativo. Sin embargo, cuando nos movemos al campo de los sistemas de recomendación basados en contenido y en conocimiento es cuando nos encontramos que las alternativas escasean. Uno de los candidatos para el desarrollo de este tipo de sistemas es jCOLIBRI⁸, un entorno de desarrollo para sistemas basados en conocimiento, principalmente aplicaciones CBR (Recio-García et al., 2008). Este entorno incluye una pequeña extensión para el desarrollo de sistemas de recomendación basados en casos, un subconjunto de los sistemas de recomendación basados en conocimiento. Aunque jCOLIBRI es un buen punto de partida para el desarrollo de sistemas de recomendación basados en conocimiento, es necesaria una ampliación de las funcionalidades que este entorno provee ya que jCOLIBRI se centra principalmente en la generación de recomendaciones basadas en

⁸<http://gaia.fdi.ucm.es/research/colibri/jcolibri>

medidas de similitud con la consulta, dejando de lado aspectos relacionados con la personalización.

En consecuencia, ninguno de estos frameworks y librerías existentes se adaptan a escenarios de recomendación como los propuestos en este trabajo de tesis, por lo tanto consideramos útil crear un framework que incluya todas las variantes del proceso de recomendación identificadas y que dé cabida al desarrollo de recomendadores basados en conocimiento en el ámbito educativo. En este sentido, nuestro trabajo supone una notable aportación en el ámbito del desarrollo de sistemas de recomendación basados en conocimiento para la enseñanza, aportando un framework en el que la creación de un sistema de recomendación se realiza de manera sencilla.

4.2. Etapas y ejes de variabilidad en el proceso de recomendación

El proceso genérico de recomendación abstraído consta de cinco etapas (obtención de la consulta, recuperación, filtrado, valoración y selección) y cada etapa puede ser abordada de distintas formas, dando lugar a los ejes de variabilidad o flexibilidad de dicho proceso. A continuación se presentan las etapas y se esbozan soluciones alternativas para ellas:

Obtención de la consulta: La estrategia básica es la formulación de la misma por parte del usuario. Aparte de este enfoque, que podríamos catalogar como *reactivo* (el sistema reacciona ante una solicitud del usuario), existe la posibilidad de iniciar el proceso de recomendación de manera *proactiva*, es decir, sin la necesidad de que el usuario introduzca información en el recomendador. En este caso es el sistema el que, en base a la información contextual, autoformularía una consulta.

Recuperación de los objetos de aprendizaje almacenados en el repositorio a partir de la consulta obtenida: Esta cuestión está relacionada con la forma en la que se indexan los objetos de aprendizaje y con los métodos que van a ser empleados para recuperar los recursos candidatos a formar parte de la recomendación. Aunque nuestros intereses actuales se centran en explotar el uso de una indexación de objetos de aprendizaje basada en el uso de ontologías, no debemos ignorar la existencia de otras alternativas de indexación. Independientemente de cuál sea la alternativa de indexación elegida, consideramos que la realización de una recuperación flexible es primordial. En este sentido, si no hay objetos de aprendizaje que satisfagan de manera estricta la consulta, se debería estudiar el uso de mecanismos que nos permitan recuperar objetos de aprendizaje indexados por un subconjunto de los (mismos o similares) términos contenidos en la consulta.

Filtrado de los objetos de aprendizaje recuperados de acuerdo a las preferencias del estudiante: El dominio del aprendizaje impone nuevas restricciones sobre el proceso de recomendación relativas a aspectos de personalización. Supongamos el caso de dos estudiantes que plantean la misma consulta al sistema. Los objetos de aprendizaje recuperados no tienen por qué ser igual de útiles para ambos, ya sea porque sus conocimientos difieren o porque tienen estilos de aprendizaje completamente distintos. En consecuencia, puede surgir la conveniencia de filtrar los objetos de aprendizaje poco útiles de acuerdo con la información contextual, teniendo en cuenta aspectos como el nivel de competencia que poseen los estudiantes, sus intereses previos, o sus estilos de aprendizaje, entre otros. Si bien este proceso de filtrado podría formar parte de la propia recuperación, considerarlo aparte permite añadir mayor flexibilidad a los enfoques de recomendación resultantes.

Valoración de la calidad de un objeto de aprendizaje: La recomendación se genera a partir de los objetos de aprendizaje más útiles para el estudiante, más similares a la consulta que la provocó, etc. La utilidad o calidad de un objeto de aprendizaje viene dada por una métrica que puede tener en cuenta algunos de los atributos del objeto de aprendizaje, la consulta y la información contextual. Una primera alternativa básica consiste en una métrica de calidad que tenga en cuenta la similitud con la consulta. Ahora bien, la métrica de calidad puede tener en cuenta otros factores, como por ejemplo la utilidad pedagógica de un objeto de aprendizaje para un estudiante o su correlación con sus intereses o estilos de aprendizaje, entre otros. Así mismo, la métrica de calidad puede enriquecerse teniendo en cuenta más de un factor, de forma que la calidad de un objeto de aprendizaje pueda obtenerse en base a una agregación de múltiples utilidades parciales. Conviene pues explorar el uso de distintas métricas de calidad, así como el modo en que estas métricas pueden ser combinadas.

Selección de los componentes de la lista de recomendación: Una recomendación suele consistir en una lista formada por el subconjunto de productos de mayor utilidad para el usuario. Si lo trasladamos a nuestro contexto, la alternativa más común consistiría en seleccionar los k candidatos más útiles de acuerdo con la valoraciones asignadas por la métrica de calidad. Esta reducción del número de candidatos es especialmente importante en nuestro contexto educativo, de modo que el alumno no se sienta agobiado al encontrarse con una carga de trabajo inabordable para la sesión de trabajo en cuestión. Sin embargo, no podemos obviar el hecho de que una lista de recomendación corta puede estar altamente sobreespecializada, de modo que los objetos de aprendizaje sugeridos se encuentren altamente correlacionados entre

ellos. Por este motivo también es interesante considerar mecanismos de selección que permitan introducir diversidad en los recursos que configuren la recomendación final.

4.3. Framework para la recomendación de objetos de aprendizaje

El estudio realizado sobre los ejes de variabilidad de un proceso genérico de recomendación basado en conocimiento como el propuesto ha dado lugar a la definición de un framework para el prototipado rápido de estos sistemas de recomendación. Una vez detectadas las etapas comunes y los ejes de variabilidad en cada una de ellas, se han definido las clases abstractas del framework encargadas de esquematizar dichas etapas, del flujo de control y que son responsables de la ejecución ordenada y específica de dichas etapas (Sección 4.3.1). A continuación se han definido los puntos flexibles o *hooks* (Pree, 1994) del framework, aquellos puntos predefinidos que necesitan ser configurados y donde han de incluirse las subclases y métodos específicos del sistema de recomendación a desarrollar (Sección 4.3.2). Por último, se han implementado un conjunto de clases que heredan de las clases abstractas definidas para este framework y que implementan algunas estrategias concretas (Sección 4.3.3). De esta forma, el desarrollo de nuevos sistemas de recomendación se vuelve más simple, tal y como veremos en la Sección 4.4.

4.3.1. Clases principales del framework

La clase principal del framework propuesto es **KBRecommender**. Esta clase contiene toda la infraestructura necesaria para implementar un sistema de recomendación de objetos de aprendizaje y es la responsable de controlar el flujo de ejecución. Todo recomendador creado a partir de este framework necesita al menos crear una subclase de **KBRecommender**, implementando los puntos flexibles que se comentarán en la Sección 4.3.2 o bien usando clases concretas ya implementadas para todos o algunos de ellos. Los parámetros que configuran el proceso de recomendación se almacenan siguiendo una arquitectura de pizarra (Stegemann et al., 2007) en una clase **Blackboard**. Este diseño permite que todas las clases encargadas del flujo de ejecución del proceso sincronicen sus resultados en la pizarra, de tal manera que todos los parámetros involucrados en el proceso de recomendación se leen y se escriben en la pizarra. Estos parámetros del proceso de recomendación se configuran en la inicialización del recomendador. Por ejemplo, un parámetro podría ser la localización de la ontología y de los objetos de aprendizaje o el número de objetos de aprendizaje que serán recomendados. Estos parámetros deben ser configurados por el propio desarrollador.

Cada una de las etapas del proceso de recomendación se corresponde con

una clase del framework:

- **QueryElicitationStage**. Se corresponde con la etapa de *Obtención de la consulta*, responsable de crear la consulta empleada en el proceso de recomendación.
- **RetrievalStage**. Se corresponde con la etapa de *Recuperación*. Ésta es responsable de, a partir de la consulta, generar un conjunto inicial de objetos de aprendizaje candidatos.
- **FilteringStage**. Se corresponde con la etapa de *Filtrado*, que se encarga de realizar una primera criba del conjunto de candidatos proveniente de la fase anterior. El comportamiento de la etapa de filtrado es común a todos los recomendadores: iterar sobre el conjunto de objetos de aprendizaje que han llegado a esta etapa y decidir para cada objeto de aprendizaje si puede pasar o no a la siguiente etapa. Esta etapa delega en un objeto de la clase **Filter** de acuerdo al patrón de diseño Estrategia o *Strategy* (Gamma et al., 1995) para decidir si un objeto de aprendizaje ha de ser eliminado del conjunto inicial recuperado.
- **RatingStage**. Se corresponde con la etapa de *Valoración*, que es la etapa responsable de asignar valores de calidad a cada uno de los objetos de aprendizaje candidatos que quedan tras la etapa de filtrado. Al igual que con la clase **FilteringStage**, el comportamiento de esta clase es común a todos los recomendadores: iterar sobre el conjunto de objetos de aprendizaje que han llegado a esta etapa, proporcionando un valor de calidad para cada uno de ellos. Del mismo modo, se ha seguido el patrón Estrategia para hacer que esta clase delegue la implementación concreta de la valoración de un objeto de aprendizaje en un objeto de la clase **QualityMetric**, de la que hablaremos más adelante.
- **SelectionStage**. Se corresponde con la etapa de *Selección*. Esta última etapa se encarga de elaborar el conjunto final de objetos de aprendizaje recomendados al estudiante.

El conjunto de clases correspondientes a las etapas de la recomendación se ve completado con las clases en las que éstas delegan y con clases que sirven para la transferencia de datos entre etapas:

- La clase **Filter** se ha diseñado pensando en que la decisión sobre si un objeto de aprendizaje debe ser eliminado de un conjunto o no sea un método flexible. Esta clase necesita que esté definido un criterio de filtrado que será utilizado por su método `bool filter(L0)`. Este método es invocado desde la clase **FilteringStage** y en función de lo devuelto por dicho método el objeto de aprendizaje correspondiente permanecerá en el conjunto o no.

- La clase **QualityMetric** se ha diseñado pensando que la valoración de un objeto de aprendizaje ha de ser un algoritmo flexible y configurable. No sólo ha de permitir que la calidad sea medida con distintas métricas de utilidad individuales sino que también es necesario poder combinar de manera flexible varias métricas para componer el valor de calidad de un objeto de aprendizaje. Para proporcionar esta flexibilidad se ha optado por emplear un patrón de composición o *Composite* (Gamma et al., 1995). De acuerdo a este patrón una **QualityMetric** puede ser un objeto de la clase **Utility**, que representa una métrica de utilidad individual, o un objeto de la clase **AggregationFunction**, que permite componer, a su vez, varios objetos de la clase **QualityMetric**. Esta composición se ha mejorado pudiendo incluir pesos para cada uno de los componentes que forman una función de agregación. La agregación de distintas métricas se realiza gracias al método `void add(QualityMetric, double)` donde el segundo parámetro indica el peso que tendrá dicha métrica de calidad en la agregación.
- Los objetos de transferencia (Alur et al., 2003) de las clases **QueryTO**, **QualityTO** y **RecommendationTO** son generados por las clases **QueryElicitationStage**, **QualityMetric** y **SelectionStage** respectivamente, para transmitir datos a las correspondientes etapas posteriores.
 - Los objetos de la clase **QueryTO** almacenan la información relativa a la consulta realizada.
 - Aunque la medida de calidad suele generar un valor normalizado en el intervalo $[0, 1]$ hemos decidido emplear un objeto de transferencia, **QualityTO**, como resultado del proceso de valoración de un objeto de aprendizaje. Este objeto de transferencia es generado por la clase **QualityMetric** y puede contener, además de dicho valor de calidad y del propio objeto de aprendizaje –representado por un objeto de la clase **LO**– todos aquellos atributos y valores que puedan ser relevantes para la siguiente fase, la etapa de selección y, por consiguiente, para el resultado final del recomendador. De esta forma aunamos en el proceso de valoración dos enfoques distintos: el habitual enfoque numérico y el criterio simbólico. Este último puede ser de utilidad, por ejemplo, a la hora de realizar justificaciones o explicaciones del comportamiento del recomendador, lo que redundará en un incremento de la confianza del usuario en el sistema.
 - Los objetos de la clase **RecommendationTO** se han generado para almacenar no sólo los objetos de aprendizaje recomendados sino también cualquier dato asociado a cada objeto de aprendizaje que sirva para arrojar información adicional sobre la recomendación generada.

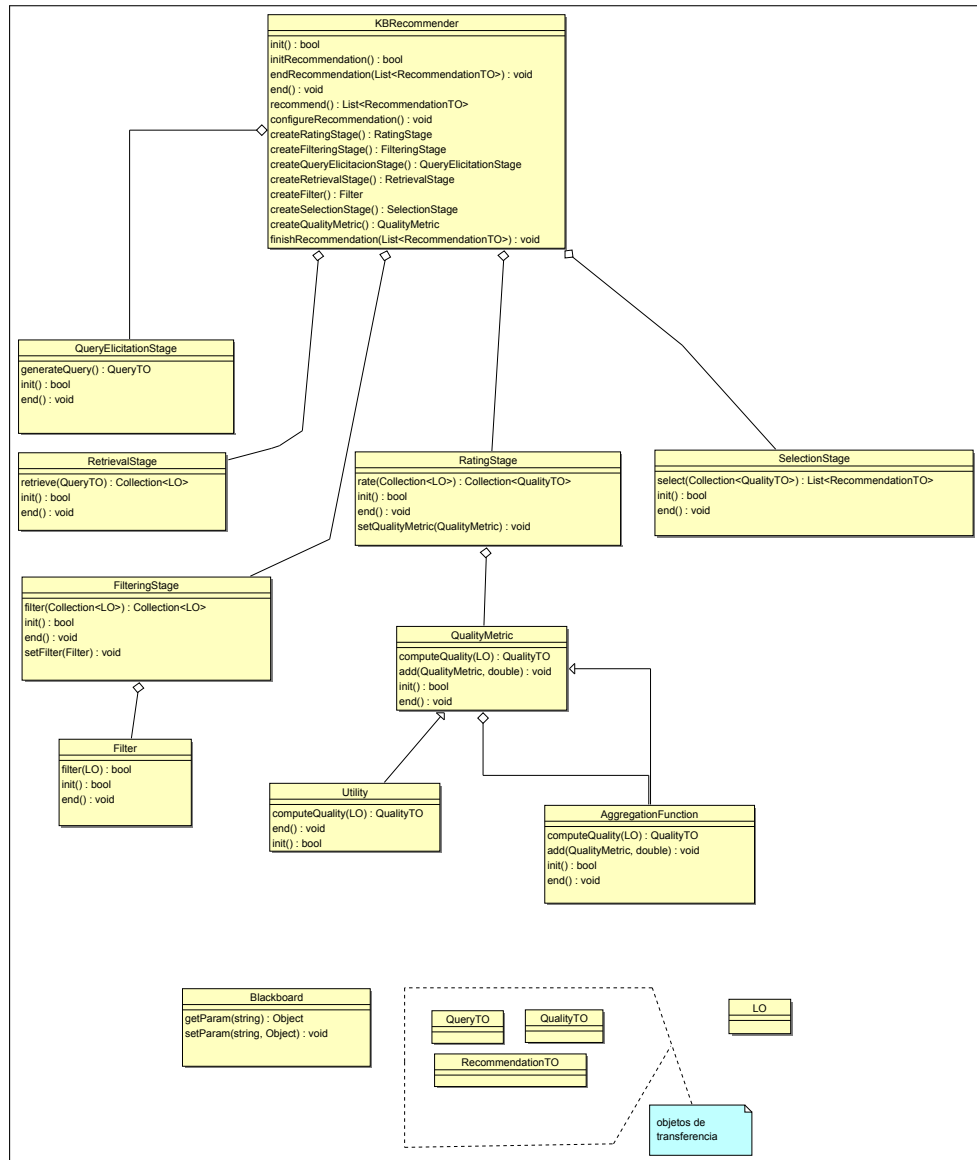


Figura 4.1: Diseño de clases que conforman el framework propuesto

La clase encargada de almacenar la pizarra, **Blackboard**, alberga el estado inicial del problema a resolver, representado por una consulta y unos parámetros de configuración del proceso de recomendación (número de objetos de aprendizaje recomendados, información contextual del estudiante, etc.). Además facilita la comunicación entre etapas almacenando información que no puede ser incluida en los objetos de transferencia. El resultado final del proceso de recomendación debe ser escrito en la pizarra como solución al problema.

Por último, como hemos adelantado, cada objeto de aprendizaje tiene una representación en un objeto de la clase **LO**. Dicho objeto guarda la información necesaria de cada objeto de aprendizaje para el proceso de recomendación, como puede ser identificador, nombre, conceptos que cubre, etc.

En términos de diseño de clases, el resultado final de lo aquí explicado aparece en la Figura 4.1.

4.3.2. Puntos flexibles del framework

Las clases descritas anteriormente presentan una serie de *hooks* o puntos flexibles que servirán para modificar el comportamiento del recomendador. Estos puntos flexibles han de ser implementados por las subclases que implementen estrategias concretas necesarias para la instanciación del framework.

Las clases que implementan las etapas del proceso de recomendación se han desarrollado usando un esquema de inicialización en dos fases. Este esquema impone que cualquier subclase de las clases abstractas que modelan las etapas de la recomendación implemente los puntos flexibles representados por los métodos `bool init()` y `void end()`. El primero se usará para validar los parámetros de configuración e inicializar la etapa, ya sea usando los parámetros obtenidos de la pizarra, ya sea usando los parámetros pasados durante la creación del objeto. El segundo se usará principalmente para liberar recursos de esta etapa y dejarla lista para que pueda ser usada durante un nuevo proceso de recomendación. Este esquema de inicialización en dos fases también lo empleamos en todas las clases en las que delegan las clases correspondientes a las fases de la recomendación. Este esquema garantiza que: (1) estos objetos puedan ser usados en múltiples ejecuciones del recomendador sin necesidad de tener que construir y destruir varias veces el mismo objeto –con la pérdida de rendimiento que esto puede producir en el sistema de recomendación–, (2) se pueda comprobar que la etapa ha sido configurada convenientemente antes de poner en marcha el recomendador –utilizando el método `bool init()` que impone el esquema– y (3) que cada etapa pueda incluir cualquier proceso adicional que deba ser ejecutado al finalizar dicha etapa –utilizando el método `void end()` que impone el esquema.

Algunas de las clases que representan las fases del recomendador también

tienen otros métodos que representan puntos flexibles:

- La clase `QueryElicitationStage` tiene como punto flexible el método `QueryTO generateQuery()`, que es responsable de implementar la creación de la consulta empleada en el proceso de recomendación.
- La clase `RetrievalStage` usa el método `Collection<LO> retrieve(QueryTO)` como punto flexible para devolver el conjunto de objetos de aprendizaje recuperados a partir de la consulta generada por la clase anterior.
- La clase `SelectionStage` tiene como punto flexible el método `List<RecommendationTO> select(Collection<QualityTO>)`. Este método genera la lista final de objetos de aprendizaje recomendados, cada uno de ellos encapsulado en un objeto `RecommendationTO`, y se construye a partir los objetos `QualityTO`, generados por la clase `RatingStage`. Este método devuelve una lista ordenada por calidad, de ahí que devuelva un objeto `List`.

Por su parte, la clase `FilteringStage` tiene un método `Collection<LO> filter(Collection<LO>)` que es el responsable de filtrar cada uno de los objetos de aprendizaje de la recuperación. Sin embargo, este método no representa un punto flexible de la etapa. Como ya adelantamos, la flexibilidad de esta clase la proporciona la delegación del comportamiento en la clase `Filter`. Esta clase sí dispone de un punto flexible, el método `bool filter(LO)`, encargado de decidir cuándo un objeto de aprendizaje debe ser filtrado.

Al igual que la anterior, la clase `RatingStage` tiene un método `Collection<QualityTO> rate(Collection<LO>)` que se encarga de valorar todos los objetos de aprendizaje pero que no representa el punto flexible de la etapa. Este método se encarga de iterar sobre los objetos de aprendizaje recuperados y filtrados y delegar en la clase `QualityMetric` para generar la valoración de un objeto de aprendizaje. Finalmente, compondrá el conjunto formado por los objetos `QualityTO` que pasarán a la siguiente fase del recomendador. La clase `QualityMetric` tiene por punto flexible al método `QualityTO computeQuality(LO)`. Tanto las métricas de utilidad individuales (`Utility`) como las funciones de agregación (`AggregationFunction`) definen su comportamiento mediante la implementación de este método, que es responsable de calcular la calidad del objeto de aprendizaje pasado como parámetro y de generar el objeto `QualityTO` asociado al mismo.

Una vez vistos los puntos flexibles relativos a las clases involucradas en el proceso de recomendación, pasamos a ver los relativos a la clase `KBRecommender`, clase principal de la que tiene que heredar cualquier recomendador implementado en este modelo. Estos puntos flexibles son los siguientes:

- `QueryElicitationStage createQueryElicitationStage()`, `RetrievalStage createRetrievalStage()`, `SelectionStage createSelectionStage()`: Estos son los métodos de factoría o *Factory Method* (Gamma et al., 1995) que se encargan de crear las instancias concretas de las fases de obtención de consulta, recuperación y selección, respectivamente. Estas fases tienen en común que las clases correspondientes son las encargadas de implementar la funcionalidad concreta de la fase sin delegar el comportamiento variable en otras clases.
- `Filter createFilter()`: Método de factoría encargado de crear el objeto de la clase `Filter` con el que la clase `FilteringStage` realiza el filtrado de los objetos de aprendizaje.
- `QualityMetric createQualityMetric()`: Este método de factoría es el responsable de crear la `QualityMetric` empleada por la `RatingStage` para valorar cada uno de los objetos de aprendizaje. En este método se crean cada uno de los objetos que compondrán la `QualityMetric`. En caso de usar una métrica de calidad con múltiples utilidades, éstas se compondrán mediante una subclase de `AggregationFunction`, usando el método `void add(QualityMetric, double)`, que permite incluir pesos para cada una de las métricas individuales que componen la métrica de calidad global. La configuración de cada una de las utilidades individuales y funciones de agregación usadas para elaborar la `QualityMetric` resultante puede emplear todos aquellos parámetros necesarios que hayan podido ser incluidos en la pizarra del recomendador durante la inicialización del mismo.
- `void configureRecommendation()`: Configura los parámetros necesarios para una recomendación, guardándolos en la pizarra. El único parámetro imprescindible para la ejecución de una recomendación es un identificador de estudiante que permitirá el acceso a la información contextual del mismo en aquellas etapas que así lo necesiten. Sin embargo, este método puede implementarse de forma que se añadan en la pizarra todos aquellos parámetros adicionales que se consideren necesarios para el recomendador concreto que se vaya a desarrollar.
- `void finishRecommendation(List<RecommendationTO>)`: Este método es el punto flexible encargado de procesar el resultado de la recomendación. Realiza, entre otras tareas, la presentación de la recomendación, almacenamiento de datos para la evaluación del recomendador o la actualización de la información contextual del usuario a partir de la recomendación proporcionada o de la interacción del usuario con la recomendación.

Por último, es necesario que exista una coordinación entre todas las etapas del proceso y sus puntos flexibles para conseguir una ejecución ordenada

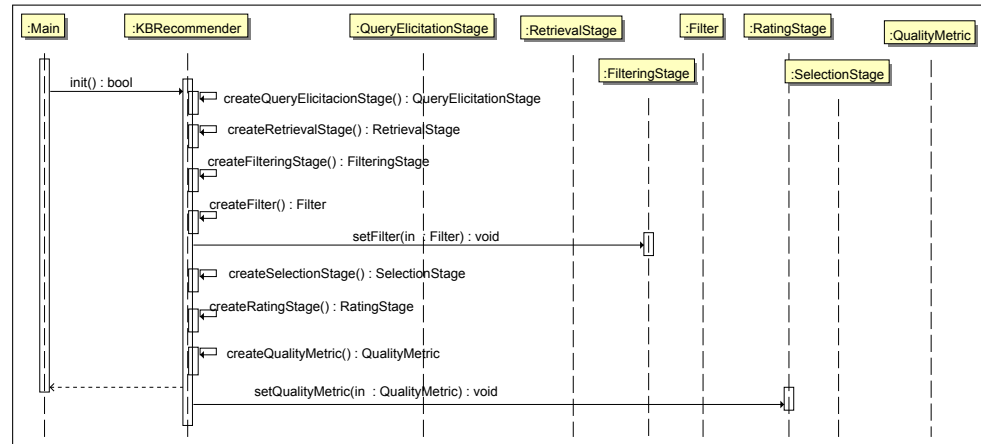


Figura 4.2: Paso de mensajes producidos por la ejecución del método `init` de la clase `KBRecommender`.

del mismo. De esto se encarga la clase `KBRecommender`, mediante los siguientes métodos de plantilla o *Template Method* (Gamma et al., 1995):

- `bool init()`: Es el método responsable de la creación de los componentes que forman el recomendador. Se encarga de la generación de las instancias que implementarán las etapas del recomendador, de la estrategia de filtrado y la métrica de la calidad usando los métodos de factoría descritos anteriormente. En la Figura 4.2 podemos ver la secuencia de paso de mensajes producida por este método.
- `bool initRecommendation()`: Este método se encarga de inicializar el recomendador para la ejecución de una recomendación. Esta tarea primero invoca el método `void configureRecommendation()`. Posteriormente se inicializan las etapas del recomendador invocando al respectivo método `bool init()` de la clase que representa cada etapa (ver Figura 4.3).
- `List<RecommendationT0>recommend()`: Es el método encargado de generar la recomendación. Ejecuta en orden cada una de las fases del recomendador, invocando el método principal de ejecución de cada una de las clases que hacen las veces de etapa del recomendador, tal y como aparece en la Figura 4.4. El resultado final es una lista ordenada de objetos `RecommendationT0` (cada uno encapsulando un objeto de aprendizaje y la información adicional asociada).
- `void endRecommendation(List<RecommendationT0>)`: Es el método responsable del procesado final de la recomendación y de la finalización de las etapas del recomendador. Esto se realiza invocando al método `void finishRecommendation(List<RecommendationT0>)`, para

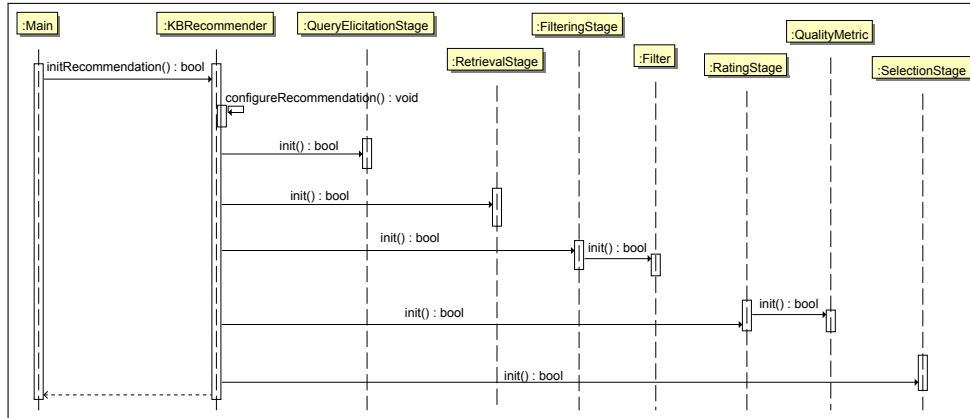


Figura 4.3: Paso de mensajes producidos por la ejecución del método `initRecommendation` de la clase `KBRecommender`.

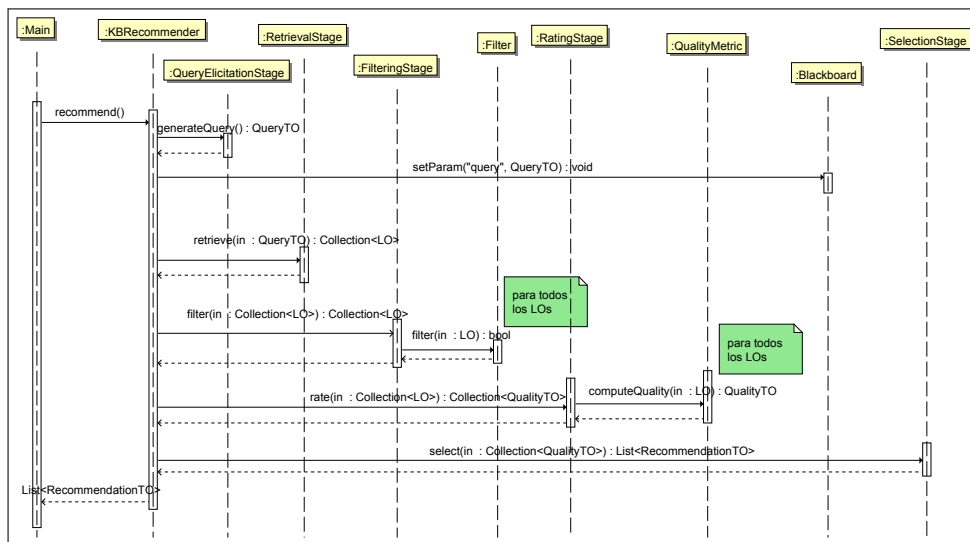


Figura 4.4: Paso de mensajes producidos por la ejecución del método `recommend` de la clase `KBRecommender`.

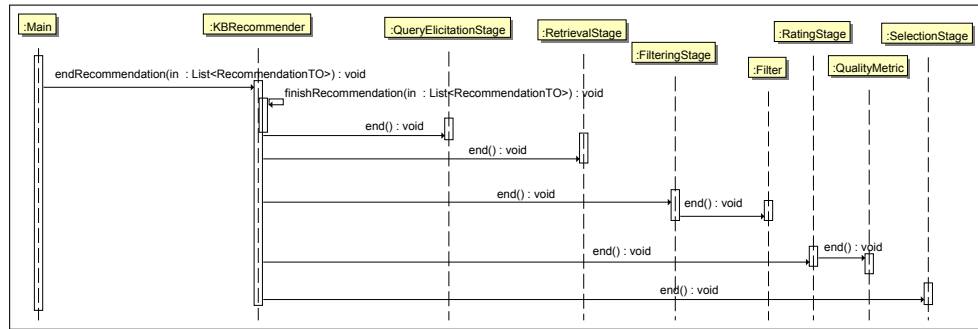


Figura 4.5: Paso de mensajes producidos por la ejecución del método `endRecommendation` de la clase `KBRecommender`.

posteriormente invocar los métodos `void end()` de cada una de las clases que representan las fases del recomendador, como se aprecia en la Figura 4.5.

- `void end()`: Es el método responsable de liberar los recursos utilizados por la etapa y dejarlos en su estado inicial.

4.3.3. Algunas implementaciones de las clases abstractas

El framework diseñado da soporte al desarrollo de sistemas de recomendación basados en distintas estrategias. Con las clases definidas hasta este momento en el framework se obliga a los desarrolladores a implementar cada una de las estrategias concretas que quieran utilizar en cada etapa del proceso de recomendación. Este diseño es lo que se conoce como framework de caja blanca (Fayad et al., 1999). Estos frameworks se caracterizan porque la instanciación de los mismos sólo es posible a través de la creación de nuevas clases. Estas clases se pueden introducir en el framework por herencia o composición. Esto implica que los desarrolladores deben comprender muy bien el framework para utilizarlo y así generar un recomendador. Sin embargo, consideramos que ésta no debe ser la única forma de uso y que deberíamos simplificar el desarrollo de estos tipos de sistemas aportando implementaciones de algunas de las estrategias con el fin de enriquecer el framework diseñado y tender hacia un modelo caja negra que facilite el desarrollo de sistemas a partir de él. Los frameworks de caja negra hacen que el desarrollo de nuevos sistemas se vuelva más simple (Fayad et al., 1999). Para crear un sistema usando un framework caja negra tan sólo se han de seleccionar los componentes predefinidos que se han implementado para cada una de las etapas y conectarlos entre sí, tal y como se verá más adelante. A continuación detallamos la implementación de varias de las clases abstractas del framework facilitando así el uso del mismo por parte de los desarrolladores.

Para la etapa de *Obtención de consulta* se han implementado las siguien-

tes clases que extienden la clase `QueryElicitationStage`. La primera es `UserQueryElicitationStage`. El método `QueryTO generateQuery()` en esta subclase se ha implementado de modo que, a través de una interfaz gráfica, interactúa con el usuario para solicitar la consulta. En esta interfaz se muestran al usuario todos los conceptos del dominio disponibles en la ontología. De entre ellos el usuario seleccionará aquel o aquellos que más le interesen, siendo éstos los conceptos que constituyan la consulta. Después, se han añadido dos clases más que generan consultas en base a la información contextual relativa al estudiante que utilice el sistema y a la información contextual relativa a la actividad. Estas clases son `ReinforcementConceptsQuery` y `DiscoverConceptsQuery`. La primera genera una consulta con un concepto que subsume a todos los conceptos ya explorados por el estudiante actual de acuerdo a la información contextual. La segunda, en cambio, genera una consulta con un concepto que subsume a aquellos conceptos que están “listos para ser explorados” para el estudiante actual. Otra implementación proporcionada es `ConceptsQueryElicitationStage`, encargada de generar una consulta que contenga un conjunto de conceptos que se obtiene como parámetro a través de la pizarra. Además se han añadido tres implementaciones pensando en extender el framework inicial con un conjunto de clases que permitan la evaluación de los sistemas de recomendación generados a partir de él. Estas clases son `FromFileQueryElicitationStage`, `CmdLineQueryElicitationStage` y `RandomQueryElicitationStage`. El método `QueryTO generateQuery()` de la primera de ellas lee de un fichero (que se establece como parámetro) las consultas presentes en el mismo y que deben ser ejecutadas por el recomendador. Por su parte `CmdLineQueryElicitationStage` genera la consulta desde los parámetros de ejecución por consola. Y por último, `RandomQueryElicitationStage` selecciona aleatoriamente un objeto de aprendizaje del repositorio y genera una consulta a partir de los conceptos cubiertos por dicho objeto de aprendizaje. En la Figura 4.6 podemos observar cómo queda el conjunto de clases definidas.

Para la etapa de *Recuperación* se han desarrollado tres estrategias distintas. La primera de ellas es una estrategia de recuperación exacta –**AccurateRetrievalStage**– encargada de recuperar de la ontología aquellos objetos de aprendizaje indexados por, al menos, todos los conceptos reflejados en la consulta. Las dos siguientes son variaciones de una estrategia de recuperación aproximada. **LightAccurateRetrievalStage** recupera del conjunto de objetos de aprendizaje aquellos que al menos tienen un concepto de los contenidos en la consulta. **ApproximateRetrievalStage** construye el conjunto de objetos de aprendizaje recuperados a partir de todos aquellos objetos de aprendizaje indexados por, al menos, un concepto de la consulta o un concepto hermano del mismo de acuerdo a su organización dentro de la ontología. Estas tres nuevas clases especializan la clase **RetrievalStage**. En la Figura 4.7 podemos observar cómo queda el conjunto de clases definidas.

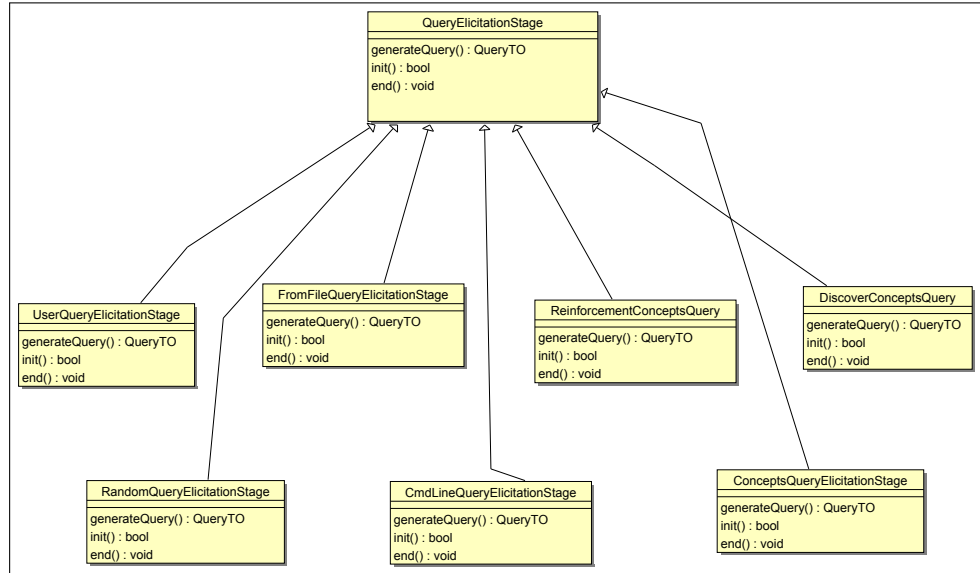


Figura 4.6: Diseño UML de las etapas de obtención de consulta proporcionadas en el framework.

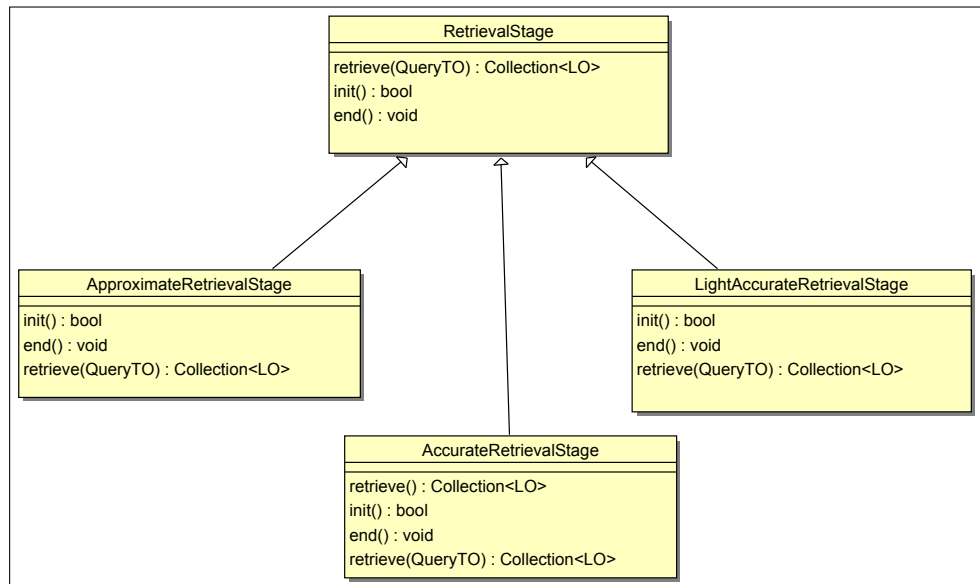


Figura 4.7: Diseño UML de las etapas de recuperación proporcionadas en el framework.

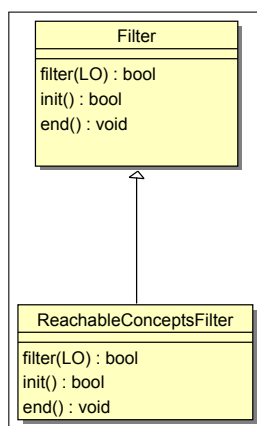


Figura 4.8: Diseño UML del filtro proporcionado en el framework

En el framework se ha incluido un filtro, como los usados en las estrategias descritas en las secciones (3.4), (3.5) y (3.6), que implementa la utilización de la información contextual de la actividad, el *itinerario de aprendizaje* definido por el instructor que marca el orden en el que se han de alcanzar los objetivos en un entorno de aprendizaje formal. Este itinerario viene definido en la ontología y los objetivos se corresponden con conceptos que el estudiante debe aprender. Con este itinerario de aprendizaje definido, un umbral de progreso (*threshold*) y la información contextual relativa al estudiante, podemos identificar qué conceptos ha alcanzado ya –conceptos *explorados*–, cuáles son los siguientes que puede estudiar –conceptos *alcanzables*– y cuáles todavía no está preparado para aprender –conceptos *no alcanzables*. A partir de esta clasificación de conceptos el filtro diseñado permite descartar, del conjunto de objetos de aprendizaje recuperados, aquellos objetos de aprendizaje que cubren conceptos no alcanzables para el estudiante dado. En el framework este filtro se ha implementado con la clase **ReachableConceptsFilter**, que extiende la clase **Filter**. La implementación del método `bool filter(LO)` consiste en comprobar, para el objeto de aprendizaje dado y el umbral de progreso definido, que el objeto de aprendizaje no cubre ningún concepto que se pueda clasificar como *no alcanzable*. En la Figura 4.8 podemos observar cómo queda la representación del filtro definido.

Para completar el framework básico se han implementado varias métricas de calidad individuales y funciones de agregación que nos permiten combinarlas. La primera de las métricas de calidad individuales implementadas valora la utilidad de un objeto de aprendizaje en función de la similitud que éste tenga con la consulta según lo definido en (González-Calero et al., 1999) y explicado en la Sección 3.4 (Ecuación 3.3). Esta métrica de calidad se ha implementado en la clase **QuerySimilarity**, que extiende la clase **Utility**.

También se ha implementado como métrica de calidad individual la utilidad pedagógica que un objeto de aprendizaje L muestra para un estudiante

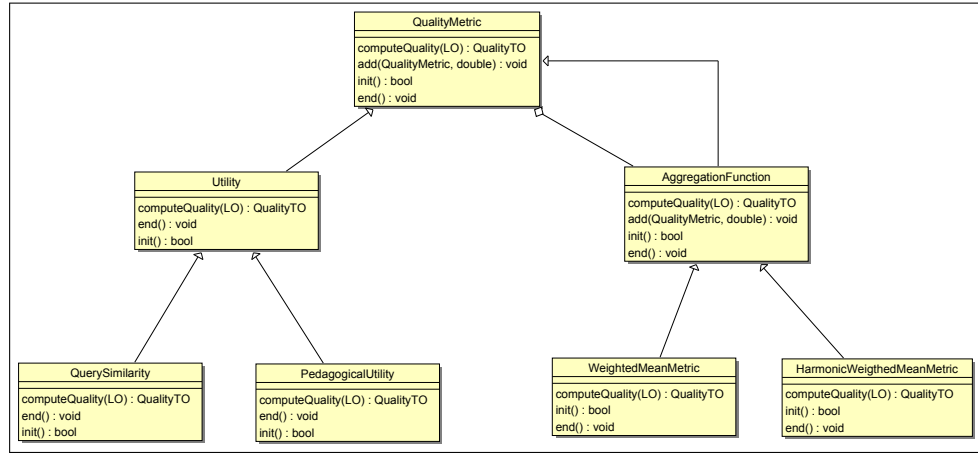


Figura 4.9: Diseño UML de las métricas de calidad que componen el framework

S dado, siguiendo la estrategia pedagógica que promueve completar las carencias de conocimiento del estudiante incluyendo conocimientos de refuerzo, que ha sido presentada en la Sección 3.4 (Ecuación 3.4). Esta métrica se ha implementado en la clase `PedagogicalUtility`, que extiende la clase `Utility`.

En cuanto a las funciones de agregación, se han incluido dos: `WeightedMeanMetric` y `HarmonicWeightedMeanMetric`. Ambas clases extienden la clase `AggregationFunction` del framework. La primera combina las utilidades como la media ponderada de todas ellas. La segunda combina las utilidades siguiendo la media armónica ponderada. En ambas funciones de agregación cuando se añade una nueva métrica de utilidad se debe indicar qué peso va a tener en la calidad final. En la Figura 4.9 podemos observar cómo queda el conjunto de clases definidas.

Tanto las métricas de utilidad individuales (`Utility`) como las funciones de agregación (`AggregationFunction`) definen su comportamiento mediante la implementación del método `QualityTO computeQuality(LO)`, que crea la información relevante sobre la calidad para el objeto de aprendizaje dado.

Finalmente, para la etapa de *Selección* se ha implementado una estrategia que selecciona los k mejores objetos de aprendizaje `TopKSelectionStage`. El método `List<RecommendationTO>select(Collection<QualityTO>)` de esta clase se encarga de ordenar el conjunto de candidatos en función de la calidad otorgada a cada objeto de aprendizaje, para después seleccionar los k primeros. El parámetro k es un número proporcionado por el desarrollador y debe ser configurado en el método `void configureRecommendation()` de `KBRecommender`. Si este número no es menor que el tamaño total del conjunto de candidatos, después de ordenar el conjunto éste se devolverá completo.

Por otro lado en el framework se han incluido también un conjunto de

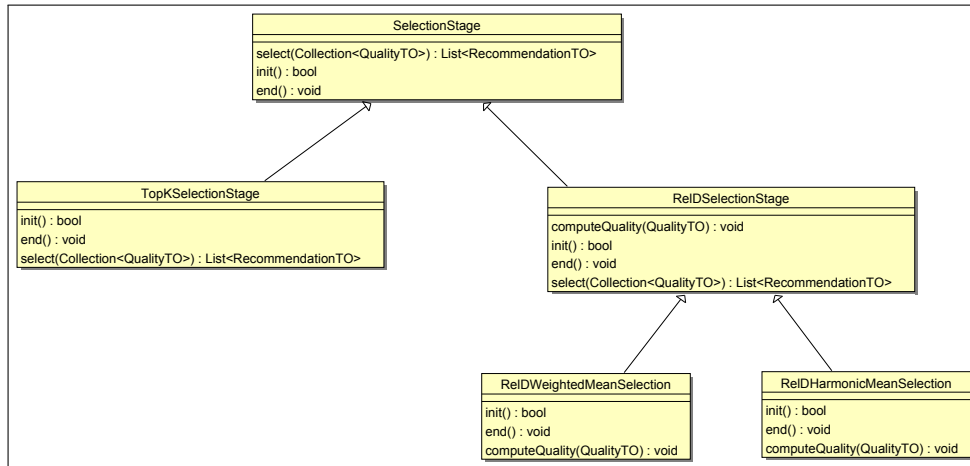


Figura 4.10: Diseño UML de las etapas de selección proporcionadas en el framework

clases `ReIDSelectionStage` y dos subclases suyas `ReIDHarmonicMeanSelection` y `ReIDWeightedMeanSelection` que implementan la estrategia de selección basada en diversidad presentada en la Sección 3.5. Sin embargo, de cara a presentar en la próxima Sección 4.4.3 cómo sería el uso del framework desde un punto de vista caja blanca, vamos a suponer que dichas clases no están incluidas y veremos cómo las implementaría un supuesto desarrollador que extiende el framework. En la Figura 4.10 podemos observar cómo queda el conjunto de clases definidas.

4.4. Instanciación de las estrategias propuestas

A continuación mostraremos con tres ejemplos cómo utilizar este framework para implementar varios prototipos de sistemas de recomendación. La implementación de un prototipo de un sistema de recomendación usando este framework es sencilla y puede realizarse de dos maneras:

- Usar las clases concretas descritas en la Sección 4.3.3 y componer el recomendador a partir de ellas.
- Si necesitamos ampliar la funcionalidad del framework, implementaremos nuevas subclases de las clases abstractas del framework y las usaremos para componer el recomendador.

De cualquier manera la utilización del framework requiere seguir los siguientes pasos:

- Crear una subclase de `KBRecommender` e implementar los puntos flexibles de esta clase descritos en la Sección 4.3.2.

- Crear una nueva clase o utilizar una clase ya existente para cada una de las etapas del recomendador. Como se ha comentado anteriormente, las etapas para las que es necesario especificar qué estrategia debe seguirse son: `QueryElicitationStage`, `RetrievalStage` y `SelectionStage`. El comportamiento de las etapas `FilteringStage` y `RatingStage` es fijo en `KBRecommender`. Si decidimos crear nuevas alternativas, para cada subclase será necesario implementar el comportamiento concreto de los métodos que se destacaron al hablar de las etapas del recomendador.
- Crear clases nuevas o utilizar las clases existentes necesarias para implementar una `QualityMetric` y crear la métrica de calidad que asignará una valoración a cada uno de los objetos de aprendizaje candidatos a ser recomendados.
- Crear una clase nueva o utilizar la clase existente necesaria para implementar `Filter` si en nuestra estrategia de recomendación vamos a incorporar algún tipo de filtrado sobre el conjunto de objetos de aprendizaje recuperados.

En esta sección vamos a detallar el proceso de creación de las tres estrategias de recomendación propuestas en este trabajo de tesis con la utilización del framework. Para mostrar la versatilidad del framework propuesto mostraremos un primer ejemplo sobre cómo construir una estrategia a partir de clases proporcionadas en el framework, es decir, éste ejemplo mostrará cómo utilizar el framework como caja negra. En un segundo ejemplo utilizaremos gran parte de las clases proporcionadas por el framework pero ampliaremos las mismas con una nueva estrategia en la etapa de selección. Esta nueva estrategia de selección se corresponde con la selección basada en diversidad y, aunque las clases que la implementan están incluidas en el framework caja negra, para el ejemplo consideramos que no están implementadas. Finalmente, el último ejemplo mostrará cómo adaptar el proceso propuesto en un modelo proactivo combinado con navegación por propuesta. Este último ejemplo muestra cómo utilizar el framework desde una perspectiva de caja blanca más acusada. Se asume que para la implementación de los tres prototipos están disponibles las tres fuentes de conocimiento necesarias para las estrategias.

4.4.1. Caso 1: Estrategia de recomendación basada en conocimiento sensible al contexto que promueve altos niveles de personalización

El primer ejemplo de prototipo de estrategia de recomendación de objetos de aprendizaje es el correspondiente a la propuesta presentada en la Sección 3.4 y en (Ruiz-Iniesta et al., 2009a,c, 2010). Recordemos que esta

estrategia opera de manera reactiva e intenta conseguir una elevada personalización usando la información contextual.

Para instanciar las etapas del recomendador hemos usado las clases que proporciona el framework:

- Para generar la consulta hemos usado la clase `UserQueryElicitationStage`.
- La etapa de recuperación elegida es `ApproximateRetrievalStage`.
- El filtro seleccionado para la etapa `FilteringStage` es `ReachableConceptsFilter`.
- Por lo que respecta a la métrica de calidad empleada en la etapa de valoración `RatingStage`, se combinarán dos utilidades individuales: la similitud con la consulta y la utilidad pedagógica del objeto de aprendizaje teniendo en cuenta la información contextual. A fin de calcular esta calidad vamos a utilizar la función de agregación implementada `WeightedMeanMetric`⁹, donde la similitud y la utilidad pedagógica se repartirán los pesos como indica el pseudocódigo que aparece al final de esta sección. La similitud se calculará utilizando la clase `QuerySimilarity`. La utilidad pedagógica se calculará mediante la clase `PedagogicalUtility`.
- Finalmente para la etapa de selección hemos utilizado la clase `TopKSelectionStage` fijando el parámetro `k=7`.

El diagrama de objetos correspondiente a la versión del prototipo explicado es el mostrado en la Figura 4.11.

Una vez que tenemos decididas las etapas, el filtro y la métrica de calidad, lo que queda por hacer es definir la clase `KBPersonalization` que hereda de `KBRecommender`, clase que modela la estrategia de recomendación y que será la encargada de controlar el flujo de la recomendación. Primero debemos asegurarnos de que el método `bool init()` de `KBPersonalization` cree las etapas de obtención de consulta, recuperación y selección deseadas. Para ello es necesario redefinir convenientemente los métodos de factoría `QueryElicitationStage createQueryElicitationStage()`, `RetrievalStage createRetrievalStage()`, y `SelectionStage createSelectionStage()` respectivamente, descritos en la Sección 4.3.2. De igual forma, para que el método `bool init()` pueda crear el filtro y la métrica de calidad deseadas debemos redefinir convenientemente los métodos de factoría `Filter createFilter()` y `QualityMetric createQualityMetric()`, respectivamente. A continuación implementaremos convenientemente los métodos responsables de la configuración de la recomendación `void configure-`

⁹El uso de otra métrica de agregación sería análogo

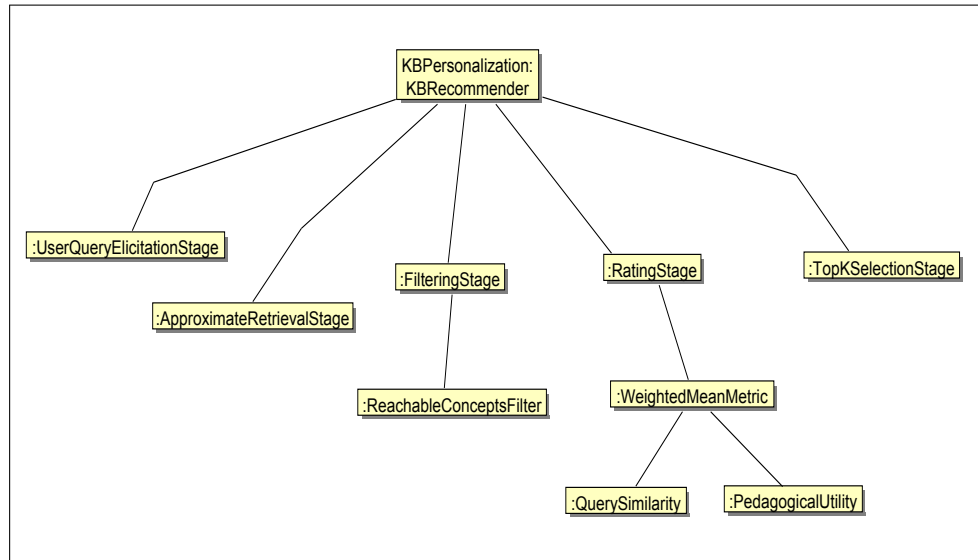


Figura 4.11: Diagrama de objetos de la estrategia de recomendación sensible al contexto que promueve altos niveles de personalización

`Recommendation()`– y de presentar al usuario la recomendación propuesta –`void finishRecommendation(List<RecommendationTO>)`. En el método `void configureRecommendation()` será necesario establecer el valor para el parámetro α –que, recordemos, sirve para ponderar la influencia que tienen la similitud con la consulta y la utilidad pedagógica en la métrica de calidad– y el umbral de progreso (*threshold*) necesario para el filtro utilizado en la estrategia. También se fija el valor del parámetro k usado por la etapa de selección y la identificación del usuario. Todos los parámetros configurados en el método `void configureRecommendation()` escriben sus valores en la pizarra. El pseudocódigo de `KBPersonalization` es el que sigue:

```

class KBPersonalization extends KBRecommender{

    void configureRecommendation(){

        Blackboard.setParam("k", 7);
        Blackboard.setParam("threshold", 5.0);
        Blackboard.setParam("alpha", 0.2);
        Blackboard.setParam("user", user);
    }

    QueryElicitacionStage createQueryElicitationStage(){
        return new UserQueryElicitationStage();
    }

    RetrievalStage createRetrievalStage(){
        return new ApproximateRetrievalStage();
    }
}

```

```

SelectionStage createSelectionStage(){
    return new TopKSelectionStage();
}

Filter createFilter() {
    return new ReachableConceptsFilter();
}

QualityMetric createQualityMetric(){
    // metrica de calidad con agregación de utilidades
    QualityMetric sim = new QuerySimilarity();
    QualityMetric pu = new PedagogicalUtility();
    QualityMetric weightedMean = new WeightedMeanMetric();
    weightedMean.add(sim, Blackboard.get("alpha");
    weightedMean.add(pu, 1-Blackboard.get("alpha");
    return weightedMean;
}

void finishRecommendation(List<RecommendationTO>){
    // mostrar al usuario la recomendación
}
}

```

4.4.2. Caso 2: Estrategia de recomendación basada en conocimiento que promueve la diversidad

Este segundo ejemplo se corresponde con la estrategia de recomendación presentada en la Sección 3.5 (Ruiz-Iniesta et al., 2011a). Como ya explicamos, esta estrategia opera de manera reactiva y su principal objetivo es aliviar la sobreespecialización en el conjunto de objetos de aprendizaje recomendados. Además incorpora la información contextual relativa a la actividad para descartar del conjunto de objetos de aprendizaje candidatos aquellos que no son adecuados para el estudiante actual.

Al igual que en el caso anterior, para realizar la instanciación de esta estrategia a partir del framework propuesto necesitaremos elegir o crear las implementaciones correspondientes para cada etapa así como un filtro y una métrica de calidad. En este prototipo haremos uso de una de las clases de obtención de consulta existentes en el framework que facilitan la realización de evaluaciones computacionales (como veremos en el Capítulo 5). Además mostraremos cómo extender el framework con una clase que realice una selección por diversidad.

Para instanciar las etapas del recomendador hemos usado las siguientes clases existentes en el framework:

- Para generar la consulta hemos usado la clase `FromFileQueryElicitationStage`.

- La etapa de recuperación elegida es `ApproximateRetrievalStage`.
- El filtro seleccionado para ser utilizado en `FilteringStage` es `ReachableConceptsFilter`.
- La métrica de calidad empleada en la etapa de valoración será la similitud con la consulta utilizando la clase `QuerySimilarity`.

Para la etapa de selección debemos definir una nueva clase que extienda `SelectionStage`. Esta nueva estrategia de selección, llamada selección por diversidad, está basada en la propuesta descrita en Smyth y McClave (2001) y explicada en la Sección 3.5. Para incluir esta nueva estrategia de selección en el framework hemos creado una nueva clase `RelDSelectionStage` como una especialización de la clase `SelectionStage` y que implementa los puntos flexibles `bool init()`, `void end()` y `List<RecommendationT0>select(Collection<QualityT0>)`. En el primero se verificará que la etapa ha sido creada con un valor válido de k (un valor mayor que cero), el método `void end()` será encargado de liberar la etapa y en el método `List<RecommendationT0>select(Collection<QualityT0>)` implementaremos el algoritmo de selección basado en diversidad.

Este algoritmo tiene la particularidad de hacer uso de una métrica de calidad formada por dos componentes, la similitud y la diversidad relativa. El primero de ellos –la similitud– es calculado y asignado a cada objeto de aprendizaje en la etapa de *valoración*. Por su parte, la diversidad relativa sólo puede ser calculada en la etapa de selección ya que su valor depende de los objetos de aprendizaje que forman el conjunto parcial de objetos recomendados en un momento concreto de la etapa de selección (ver Ecuación 3.7). El cálculo de la calidad se realizará en el método `void computeQuality(QualityT0)` definido en `RelDSelectionStage`. Para otorgar flexibilidad al cálculo de la calidad en este algoritmo de selección y que pueda seguir distintas aproximaciones se ha utilizado el patrón *Template Method* para la implementación. Este patrón permite que partes del algoritmo de cálculo definido en `RelDSelectionStage` pueda ser redefinido en todas aquellas clases que la extiendan. La clase `RelDSelectionStage` define el algoritmo de selección basado en diversidad dejando como un método abstracto el cálculo de la calidad (`void computeQuality(QualityT0)`) que será redefinido en sus subclases (véase en la Figura 4.10 la jerarquía de clases que ha quedado finalmente definida en el framework).

Para este prototipo se ha desarrollado la clase `RelDHarmonicMeanSelection`¹⁰. Esta clase redefine el método `void computeQuality(QualityT0)` de tal manera que la calidad para un determinado objeto de aprendizaje es la media armónica ponderada de la similitud con la consulta de

¹⁰En el framework también se ha incluido la clase `RelDWeightedMeanSelection` que implementa en el método `void computeQuality(QualityT0)` el cálculo de la calidad como la media ponderada

dicho objeto de aprendizaje (calculada en la etapa de valoración) y la diversidad relativa previamente calculada dentro del método `List<RecommendationTO>select(Collection<QualityTO>)`. Además esta clase necesita un parámetro `weight` que indica qué peso se le otorga a cada factor en la ponderación (debe ser configurado en el método `void configureRecommendation()` de la subclase `KBdiversity` de `KBRecommender` que representa al recomendador). Mostramos a continuación el pseudocódigo de la clase que implementa la selección por diversidad y la especialización del método `void computeQuality(QualityTO)` en la clase `RelDHarmonicMeanSelection`. El diagrama de objetos correspondiente a este recomendador es el mostrado en la Figura 4.12.

```
class RelDSelectionStage extends SelectionStage{

    public List<RecommendationTO> select(Collection<QualityTO>
        theList) {
        p = new List<RecommendationTO>();
        //repetir hasta que el tamaño de p sea k
        //para cada QualityTO q contenido en theList
            // Calcular relD de q con respecto a p
            // Guardar relD en q
            computeQuality (q)
        // Ordenar theList en función de la calidad computada
        // para cada uno de sus elementos
        // insertar en p el primer elemento de theList
        // eliminar el primer elemento de theList

        return p
    }

    protected abstract void computeQuality(QualityTO q);

    public bool init() {
        // comprobar que k es mayor que cero
    }

    public void end() {
        // liberar la etapa
    }
}
```

```
class RelDHarmonicMeanSelection extends RelDSelectionStage {

    public void computeQuality (QualityTO q) {
        // Obtener similitud de q;
        // Obtener relD de q;
        // Guardar ((similitud *relD)/((Blackboard.getParam("weight"
            ") *relD)+(1-Blackboard.getParam("weight")) *similitud))
            ) en q;
    }
}
```

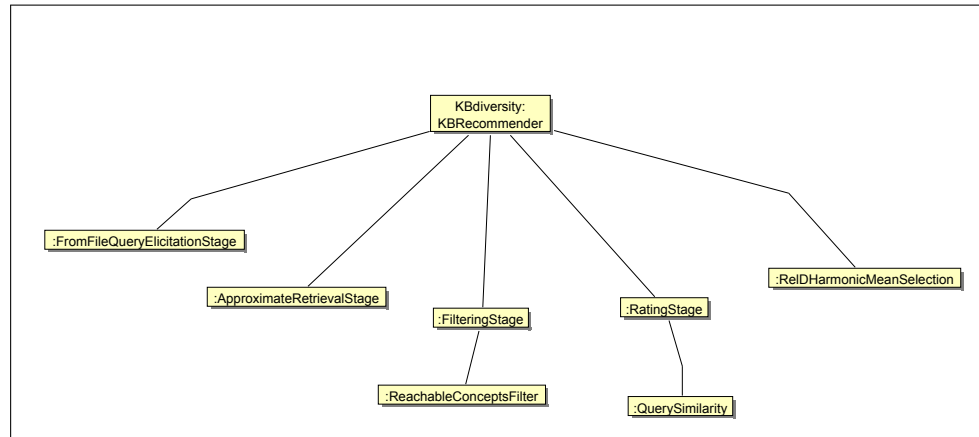


Figura 4.12: Diagrama de objetos de la estrategia que promueve la diversidad en el conjunto de recomendados

Después de elegir o crear todas las etapas, así como el filtro y la métrica de calidad, definimos la clase **KBdiversity** que modela el recomendador y que será la encargada de controlar el flujo de la recomendación. En un primer momento será necesario asegurarse de que el método `bool init()` de **KBdiversity** cree las etapas de Obtención de consulta, Recuperación y Selección deseadas. Por ello hay que redefinir convenientemente los métodos de factoría **QueryElicitacionStage** `createQueryElicitationStage()`, **RetrievalStage** `createRetrievalStage()` y **SelectionStage** `createSelectionStage()`, respectivamente descritos en la sección 4.3.2. Del mismo modo, es necesario redefinir convenientemente los métodos de factoría **Filter** `createFilter()` y **QualityMetric** `createQualityMetric()` para que el método `bool init()` de **KBdiversity** cree el filtro y la métrica de calidad a usar en la etapa de filtrado y en la etapa de valoración, respectivamente. A continuación implementaremos convenientemente los métodos responsables de la configuración de la recomendación –`void configureRecommendation()`– y de presentar al usuario la recomendación propuesta –`void finishRecommendation(List<RecommendationT0>)`. El pseudocódigo de esta clase es el que sigue:

```

class KBdiversity extends KBRecommender{

    void configureRecommendation(){
        Blackboard.setParam("k", 7);
        Blackboard.setParam("threshold",5.0);
        Blackboard.setParam("user", user);
        Blackboard.setParam("weight", 0.1);
    }

    QueryElicitacionStage createQueryElicitationStage(){
        return new FromFileQueryElicitationStage();
    }
}
  
```

```

    }

    RetrievalStage createRetrievalStage(){
        return new ApproximateRetrievalStage();
    }

    SelectionStage createSelectionStage(){
        return new RelDHarmonicMeanSelection();
    }

    Filter createFilter() {
        return new ReachableConceptsFilter();
    }

    QualityMetric createQualityMetric(){
        return new QuerySimilarity();
    }

    void finishRecommendation(List<RecommendationTO>){
        // realizar los cálculos para la evaluación computacional
    }
}

```

4.4.3. Caso 3: Estrategia de recomendación basada en conocimiento que combina proactividad y navegación por propuesta

Este último ejemplo muestra el proceso de realización del prototipo para la estrategia de recomendación que usa la navegación por propuesta como mecanismo de interacción con el usuario (Ruiz-Iniesta et al., 2009b).

Si recordamos, esta estrategia presenta al usuario k objetos de aprendizaje como primera recomendación. Para cada objeto el usuario tiene la opción de visualizarlo o mostrar sus preferencias (seleccionando un objeto de aprendizaje y diciendo que quiere “más como éste”) para refinar la propuesta y obtener una nueva recomendación. Si se da la primera opción llegamos al final del proceso de recomendación. Si, por el contrario, el alumno proporciona una preferencia para refinar la recomendación, el recomendador propondrá de nuevo otro conjunto de objetos de aprendizaje al estudiante como resultado del refinamiento del conjunto anterior. Este proceso se repite siguiendo el modelo de navegación explicado en la Sección 3.6.

La novedad de esta estrategia con respecto a las dos anteriores radica en la manera de interacción con el usuario y en el mecanismo de generación de las consultas. Como explicaremos a continuación, lo que necesita esta estrategia es una nueva clase que se encargue, entre otras cosas, de encadenar distintos procesos de recomendación para así conseguir esa navegación por propuesta. Más concretamente, para poder usar el framework en esta nueva manera de interacción con el usuario, será necesario crear una clase que se encargue

de recoger la información contextual del estudiante, a continuación a partir de dicha información generará las consultas que deba ejecutar el proceso de recomendación y escribirlas en la pizarra. Esta nueva clase que implementa y gestiona la navegación es **Navigation**.

La clase **Navigation** tiene los siguientes métodos:

- **void transit()**: Este método es el encargado de actualizar el valor del parámetro que indica en que fase de la navegación estamos y, si es el caso, recoger las preferencias mostradas por el estudiante en la propuesta (ver Sección 3.6).
- **Collection<QueryT0> generateQueries()**: Este método es el encargado de generar tantas consultas como sea necesario según en la fase en la que se encuentre la estrategia. Se generarán tantas consultas como número de grupos en los que deba dividirse el repositorio. La generación de consultas se hará mediante las clases existentes en el framework, **DiscoverConceptsQuery**, **ReinforcementConceptsQuery** y **ConceptsQueryElicitationStage**, usando la que corresponda conforme a la fase de la estrategia en la que se encuentre la navegación. En el caso de esta última, su método **QueryT0 generateQuery()** debe ser llamado tantas veces como número de grupos deba dividirse el repositorio. Antes de cada llamada debe almacenarse en la pizarra el conjunto de conceptos que constituye la consulta.
- **Map<QueryT0,List<RecommendationT0>> execute(Collection<QueryT0>)**: Este método es el encargado de ejecutar las restantes etapas del proceso de recomendación para cada consulta generada. Recibe como parámetro la colección de consultas generadas por el método anterior y devuelve un **Map**, una estructura que empareja cada objeto **QueryT0** con la recomendación producida (**List<RecommendationT0>**).
- **List<RecommendationT0> greedySelection(Map<QueryT0,List<RecommendationT0>>,int)**: Este método es el encargado de seleccionar los k mejores objetos de aprendizaje que serán mostrados al estudiante. Recibe como parámetros una estructura de pares consulta-recomendación generada por el método **Map<QueryT0, List<RecommendationT0>> execute(Collection<QueryT0>)** y el parámetro k que indica cuántos objetos de aprendizaje deben ser seleccionados. La selección se realiza con un método voraz. Este método primeramente selecciona el mejor objeto de aprendizaje de cada una de las listas de objetos de aprendizaje recuperados. Con esto tenemos n objetos de aprendizaje que formarán el conjunto de recomendados. Si n es menor que k entonces se completa el conjunto de recomendados con los restantes $n - k$ mejores representantes.

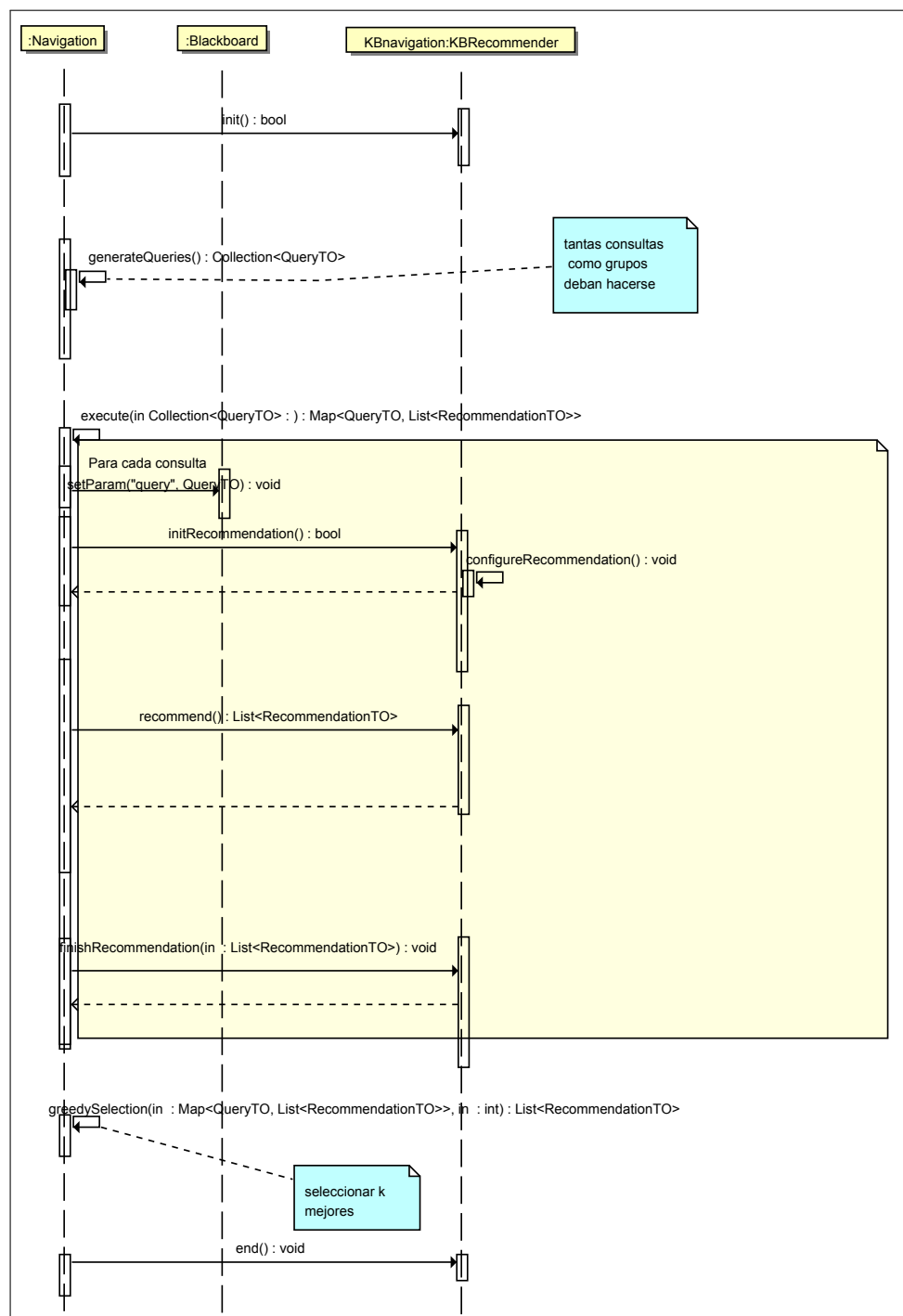


Figura 4.13: Paso de mensajes producido por la ejecución de una recomendación en *Navigation*

En la Figura 4.13 podemos observar un diagrama del paso de mensajes producido por la clase `Navigation` para generar una recomendación en `KBnavigation` después de haber obtenido las preferencias del estudiante, es decir, nos encontramos en la segunda fase de la estrategia después de haberse ejecutado el método `void transit()`. Es importante señalar que el método `bool initRecommendation()` ejecutado al comienzo del método `Map<QueryT0,List<RecommendationT0>> execute(Collection<QueryT0>)` sólo inicializará aquellas etapas del proceso que estén definidas. Para el caso concreto de este recomendador la etapa de obtención de consulta es una etapa no definida y no será inicializada ni ejecutada.

Una vez definida esta nueva clase pasamos a definir cuáles son las clases necesarias para instanciar las etapas del proceso de recomendación.

Para realizar la instanciación de esta estrategia a partir del framework propuesto, necesitaremos elegir las implementaciones correspondientes para las etapas de recuperación, filtrado, valoración y selección así como un filtro y una métrica de calidad.

- Como etapa de recuperación se ha seleccionado `LightAccurateRetrievalStage`.
- El filtro elegido es `ReachableConceptsFilter`.
- La métrica de calidad elegida es `WeightedMeanMetric` que, como en el primer ejemplo mostrado, combinará dos utilidades individuales: la similitud con la consulta y la utilidad pedagógica del objeto de aprendizaje teniendo en cuenta la información contextual.
- Por último la selección de los objetos de aprendizaje en cada proceso de recomendación se hará utilizando la clase `TopKSelectionStage`.

En la Figura 4.14 vemos el diagrama de objetos de la composición de las clases que forma el recomendador `KBnavigation`.

Finalmente mostramos el pseudocódigo del recomendador utilizado en cada iteración del proceso de navegación de este último prototipo. La implementación del método `QueryElicitationStage createQueryElicitationStage()` que vemos en el pseudocódigo siguiente hace que el método `bool init()` heredado de `KBRecommender` no genere la etapa de obtención de consulta. Consecuentemente, se consigue que el método `bool initRecommendation()` heredado también de `KBRecommender` funcione bien, porque no inicializa la obtención de consulta al no haberse generado consulta alguna.

```
class KBnavigation extends KBRecommender{

    void configureRecommendation(){
        Blackboard.setParam("k", 7);
        Blackboard.setParam("threshold",5.0);
        Blackboard.setParam("user", user);
    }
}
```

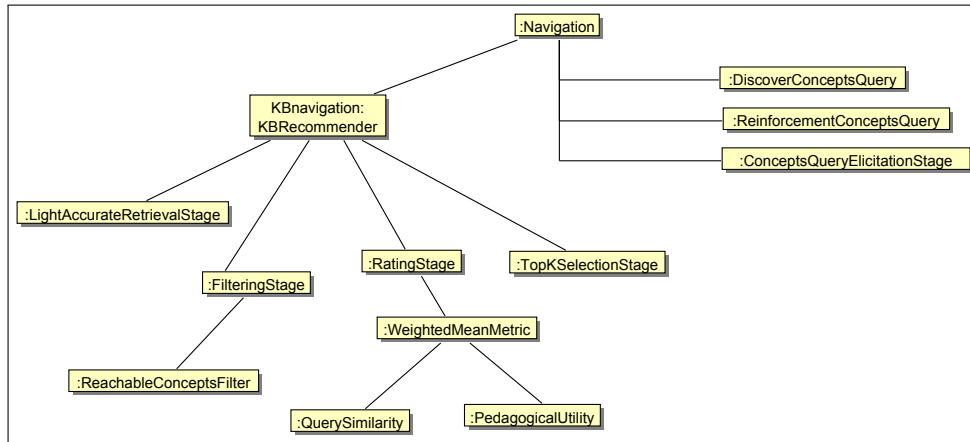


Figura 4.14: Diagrama de objetos de la estrategia que utiliza un modelo de navegación por propuesta

```

    Blackboard.setParam("alpha", 0.2);
}

QueryElicitationStage createQueryElicitationStage(){
    return null;
}

RetrievalStage createRetrievalStage(){
    return new LightAccurateRetrievalStage();
}

SelectionStage createSelectionStage(){
    return new TopKSelectionStage();
}

Filter createFilter() {
    return new ReachableConceptsFilter();
}

QualityMetric createQualityMetric(){
    QualityMetric sim = new QuerySimilarity();
    QualityMetric pu = new PedagogicalUtility();
    QualityMetric weightedMean = new WeightedMeanMetric();
    weightedMean.add(sim, Blackboard.get("alpha");
    weightedMean.add(pu, 1-Blackboard.get("alpha");
    return weightedMean;
}

void finishRecommendation(List<RecommendationTO>){
    // no es necesario hacer nada
}
}

```

4.5. Conclusiones

En este capítulo se ha presentado un framework que permite el prototipado rápido de sistemas de recomendación basados en conocimiento para la recomendación de objetos de aprendizaje. Este framework supone una aportación al campo del desarrollo de sistemas de recomendación basados en conocimiento: no existían herramientas centradas en el diseño de este tipo de sistemas en el dominio de la enseñanza y además introduce un proceso de recomendación basado en etapas que permite la configuración de múltiples sistemas de recomendación de manera sencilla.

Para el diseño del framework partimos del análisis de un proceso de recomendación en el que se han identificado cinco etapas: obtención de la consulta, recuperación, filtrado, valoración y selección. Cada una de estas etapas es considerada como un eje de variabilidad del proceso de recomendación. Esto supone que se pueden configurar distintos procesos de recomendación haciendo variar alguna de las etapas.

El framework ha sido diseñado de manera que se puedan crear fácilmente recomendadores que usen diversas estrategias ya implementadas para cada una de las etapas. Asimismo, el framework puede extenderse fácilmente implementando nuevas estrategias. Para ello se deben crear clases concretas de las etapas deseadas y configurar después el recomendador deseado.

Finalmente en el capítulo se ejemplifica el uso del framework con la construcción de las tres estrategias de recomendación que componen el trabajo central de esta tesis. La primera estrategia mostrada hace uso de las implementaciones proporcionadas en el framework. El segundo caso muestra la instanciación de la estrategia basada en diversidad. Para esta estrategia se ha simulado la creación de una nueva clase que modela el comportamiento del algoritmo de diversidad deseado (realmente la clase se encuentra incorporada en el framework). El último recomendador implementado muestra la estrategia de recomendación que incluye como modelo de interacción con el usuario la combinación de proactividad y navegación por propuesta. En este último ejemplo se muestra cómo ha sido el proceso de integración del módulo de navegación por propuesta en el proceso de recomendación, y con ello se muestra una de las características del framework, poder utilizar distintas configuraciones del proceso de recomendación en un mismo recomendador, mediante la modificación de la etapa de generación de consulta en las distintas fases del recomendador.

Para finalizar, señalar que el framework desarrollado nos ha ayudado a construir, de manera rápida, distintos sistemas de recomendación que hemos utilizado en las evaluaciones de las estrategias de recomendación que se detallan en el próximo capítulo.

Capítulo 5

Evaluación de las estrategias de recomendación

*Por muy hermosa que sea la estrategia,
de vez en cuando debes mirar los
resultados.*

W. Churchill

La evaluación de los sistemas de recomendación ha sido, y sigue siendo, objeto de constante investigación. La evaluación de estos sistemas se convierte en un punto crítico para la mejora y éxito de los mismos. Tradicionalmente, la evaluación de los sistemas de recomendación se ha centrado en medir la precisión del sistema en la predicción de las preferencias del usuario, en otras palabras, las valoraciones estimadas se comparan con las valoraciones reales y las diferencias entre ellas se calculan mediante medias de errores (Herlocker et al., 2004). En términos de la utilidad de las recomendaciones para un usuario existe, sin embargo, una creciente aceptación de que la calidad de los elementos recomendados puede ser más importante que la exactitud en la predicción de las valoraciones (Shani y Gunawardana, 2011). Como resultado de esto, las métricas que evalúan la calidad de los n mejores elementos de la lista de recomendación están siendo ampliamente consideradas en los trabajos de evaluación.

Las estrategias propuestas en este trabajo de tesis tienen la responsabilidad de recomendar recursos de aprendizaje a estudiantes. En este caso, es más importante analizar la calidad de estos recursos por el posible impacto que tendrán en el aprendizaje del estudiante. Por lo tanto, es necesario hacer uso de métricas de evaluación que permitan analizar la calidad de las listas de recomendación, en base a la calidad de los elementos contenidos en ellas, para así poder observar el comportamiento de las mismas. En esta tesis se analizará el comportamiento de las estrategias propuestas desde tres características de las listas de recomendación: la utilidad pedagógica, la similitud

de los recursos contenidos en ella y la diversidad de los elementos de la lista.

Sin embargo, no podemos olvidar que las estrategias aquí presentadas están diseñadas para la utilización en un entorno de aprendizaje. Por lo tanto es necesario analizar la aceptación de las estrategias de recomendación por parte de los usuarios así como la satisfacción de los mismos sobre el uso. Para ello se ha llevado a cabo una evaluación directa mediante encuestas realizadas a un grupo de profesores y varios grupos de estudiantes que utilizaron las estrategias durante un tiempo determinado.

Para poder llevar a cabo ambas evaluaciones será necesario aplicar las estrategias propuestas a un dominio educativo y un repositorio concretos. En nuestro caso hemos elegido un repositorio de objetos de aprendizaje de Programación. Comenzaremos presentando el repositorio de partida y cuál ha sido su evolución hasta llegar al estado actual, y detallaremos las fuentes de conocimiento concretas usadas en esta ejemplificación de las estrategias (Sección 5.1). Una vez establecidas las fuentes de conocimiento y usando los prototipos desarrollados con ayuda del framework descrito en el capítulo anterior, en la Sección 5.2 presentaremos un análisis del comportamiento de las estrategias y en la Sección 5.3 presentaremos la evaluación realizada con estudiantes y profesores. Finalizaremos el capítulo (Sección 5.4) con unas pequeñas conclusiones.

5.1. Aplicación de las estrategias de recomendación a un repositorio de recursos educativos de Programación

Las estrategias presentadas en este trabajo de tesis han sido aplicadas en un escenario formal de educación, donde se dispone de un repositorio de objetos de aprendizaje de Programación. Este repositorio ha sido desarrollado siguiendo un modelo productor-consumidor (Downes, 2007) por un grupo de profesores de la Facultad de Informática de la Universidad Complutense de Madrid. A lo largo del tiempo, el repositorio y la manera de acceder a los recursos ha evolucionado (Gómez-Albarrán et al., 2011). En un primer momento los recursos (ejercicios resueltos) estaban accesibles para que los alumnos navegaran libremente por el repositorio explorando los contenidos que quisieran. Después, el equipo docente involucrado introdujo una importante mejora didáctica: incorporar material práctico no resuelto, acompañado en ocasiones por guías para su resolución y la comprobación de su correcto funcionamiento. No obstante, esta ampliación de contenidos no fue suficiente para facilitar el auto-aprendizaje. El acceso al repositorio evolucionó para implementar un curso con un modelo secuencial de aprendizaje en el que los materiales se repartían por temas y para avanzar de un tema a otro era necesario superar un test. Después de esta segunda mejora se realizó una

encuesta a los estudiantes involucrados en el uso del repositorio y, aunque los estudiantes valoraron la ayuda que estos recursos les proporcionan, cerca del 70 % de los estudiantes reconocieron que echaban en falta facilidades más sofisticadas para el acceso a los mismos (Gómez-Albarrán et al., 2011). Esto ha llevado a mejorar el repositorio añadiendo una nueva manera de acceso al mismo que hiciera uso de las estrategias de recomendación propuestas.

En la actualidad se cuenta con un repositorio de objetos de aprendizaje, en continuo crecimiento, que alberga más de 500 recursos educativos de entre los siguientes tipos:

- Ejercicios *resueltos*: ejemplos de ejercicios resueltos para que el estudiante pueda analizar las soluciones.
- Ejercicios de *desarrollo completo*: a partir del enunciado de un problema, el estudiante debe proponer una solución al mismo.
- Ejercicios *fill in the gaps*: ejercicios para completar.
- Ejercicios *encuentra el error*: ejercicios con algún error que el estudiante debe ser capaz de descubrir y solucionar.
- Cuestiones: breves preguntas para auto-evaluar conocimientos.

A continuación describiremos el proceso de aplicación de las estrategias de recomendación generales propuestas a este repositorio de objetos de aprendizaje para la enseñanza de la Programación en lo que respecta a las necesidades de conocimiento: los objetos de aprendizaje, la ontología y la información contextual.

5.1.1. Los objetos de aprendizaje

Parte del trabajo de aplicación de las estrategias de recomendación al repositorio de partida ha implicado realizar una adaptación de los recursos educativos existentes en el repositorio al estándar LOM. Para ello hemos utilizado el lenguaje XML con el fin de marcar con metadatos los distintos objetos de aprendizaje contenidos en el repositorio. Estos metadatos nos permitirán generar el conocimiento necesario sobre los distintos objetos de aprendizaje que se verá reflejado en la ontología y será utilizado posteriormente para razonar sobre ellos. El marcado de estos recursos se ha realizado, como ya se indicó en el Capítulo 3, utilizando en el más alto nivel las siguientes categorías propuestas en LOM: *general*, *ciclo de vida*, *técnica*, *educativa* y *relación*.

Para facilitar el proceso de adaptación de estos recursos se ha desarrollado una herramienta auxiliar llamada LOEditor. Esta herramienta está pensada como una herramienta de autoría de objetos de aprendizaje (Ruiz-Iniesta

et al., 2012b). Dicha herramienta permite etiquetar un documento (en cualquier formato digital) con aquellos metadatos necesarios para su correcta adaptación al modelo de representación aquí necesitado. El usuario selecciona el fichero con el recurso que quiere marcar con metadatos, a continuación indica el título del recurso, el autor (o autores), un pequeño resumen del contenido y finalmente seleccionará de una lista las *keywords* que representarán los conceptos del dominio que cubre el recurso. Además deberá introducir el resto de información necesaria: el tipo de recurso que es (ejercicio resuelto, ejemplo, etc.), el nivel de dificultad, si está relacionado con algún otro objeto de aprendizaje del repositorio, etc. Finalmente, la herramienta automáticamente detecta la información técnica del recurso, genera el documento LOM asociado y lo añade al repositorio.

5.1.2. La ontología del dominio

La ontología nos permite dotar al sistema de conocimiento sobre el dominio concreto de aplicación, conocimiento que, como vimos en el capítulo anterior, juega un papel importante en las estrategias de recomendación definidas. Para esta aplicación de las estrategias de recomendación, el conocimiento de la ontología será relativo al dominio de la Programación. El desarrollo de la ontología ha seguido la metodología METHONTOLOGY (Fernández-López et al., 1997) y nos hemos basado en otras ontologías existentes sobre lenguajes de programación (Sosnovsky y Gavrilova, 2006). La ontología ha sido desarrollada en OWL mediante la herramienta Protégé¹.

La ontología desarrollada tiene una clase principal, **ProgrammingConcept**, que aglutina los conceptos relativos a la Programación. Bajo esta clase está organizada la taxonomía de conceptos. La composición final de la jerarquía contiene 34 clases que representan conceptos en distintos niveles de la jerarquía y 28 instancias que representan elementos concretos en los lenguajes de programación C++ y Pascal. La Figura 5.1 muestra un fragmento de la ontología. En ella podemos ver algunas de las 34 clases definidas. Por ejemplo, hemos definido la clase **Conditional** que representa a las estructuras condicionales y **switch-sentence** e **if-sentence** son instancias de dicha clase y representan las correspondientes estructuras en el lenguaje de programación C++.

La ontología además establece un vínculo entre cada objeto de aprendizaje contenido en el repositorio y los conceptos del dominio que éste cubre. Esto es necesario para poder asociar cada objeto de aprendizaje a los conceptos que éste permite aprender y que así las estrategias de recomendación puedan razonar sobre los objetos de aprendizaje en términos de los conceptos que cada uno de ellos cubre. Cada objeto de aprendizaje contenido en el repositorio será representado como una instancia de la clase **LearningObject**. Para

¹<http://protege.stanford.edu/>

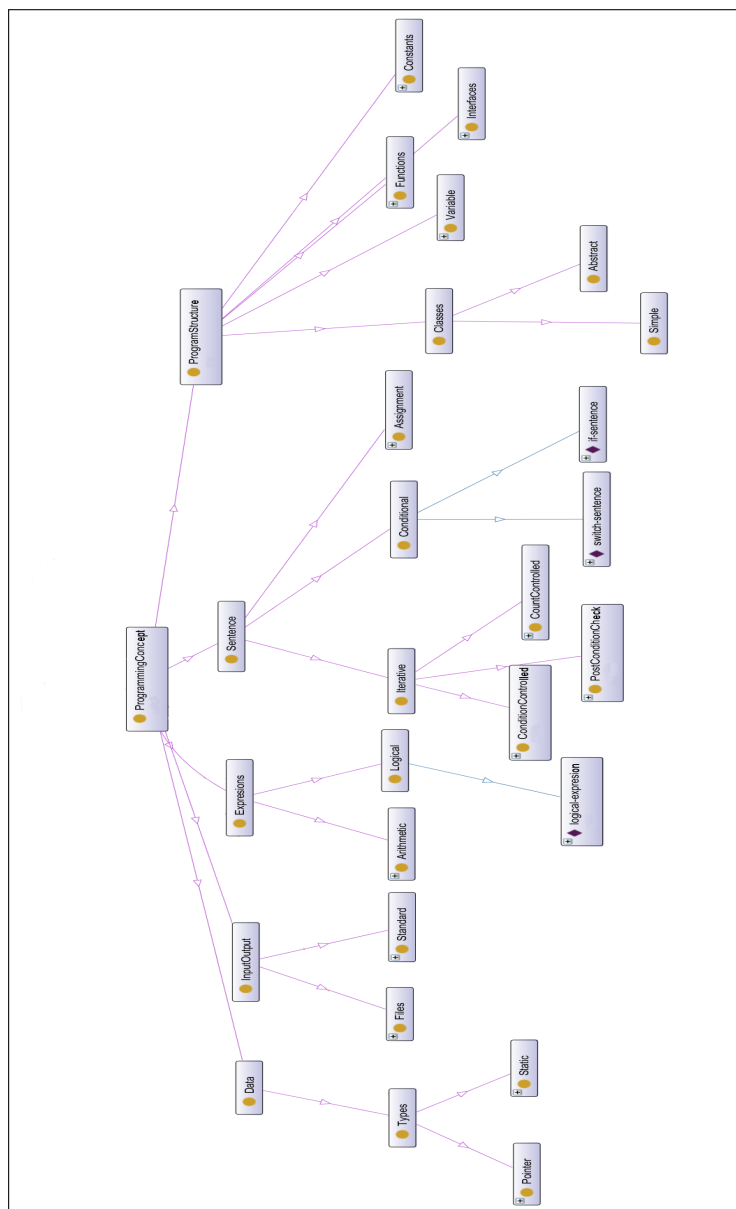


Figura 5.1: Vista parcial de la ontología del dominio

facilitar el trabajo de incluir esta representación de los objetos de aprendizaje en la ontología se ha desarrollado la herramienta LOM2OWL, que a partir del fichero XML que representa a un objeto de aprendizaje lo transforma en una instancia de **LearningObject** con las propiedades necesarias. Esta herramienta ha sido desarrollada en Java y hace uso de la librería **OntoBridge**². Más concretamente, esta herramienta analiza el metadato **keyword** de cada objeto de aprendizaje contenido en el repositorio, que como hemos dicho indica qué conceptos de los definidos en la ontología del dominio son cubiertos por ese objeto de aprendizaje. A continuación esta herramienta genera una nueva instancia en la ontología que representa dicho objeto de aprendizaje y completa la propiedad **covers** de dicha instancia con las instancias que representan a los elementos del dominio identificados.

5.1.3. La información contextual

La información contextual como vimos en la Sección 3.3.3 está formada por dos elementos: contexto de la actividad y contexto del estudiante.

El contexto de la actividad es el itinerario de aprendizaje. Por ejemplo, para el uso que en un escenario como el nuestro se va a dar a los recomendadores desarrollados, se ha considerado que el concepto **Iterative** (los bucles) debería ser aprendido después del concepto **Conditional** (los condicionales), y antes del concepto **Functions** (subprogramas). Esta información contextual se ha representado en la ontología gracias a dos propiedades que definen el secuenciamiento entre instancias de conceptos **has_next** y **has_previous**. Estas propiedades nos indican el concepto que precede y sigue, respectivamente, al concepto actual. Usando estas dos propiedades se podrá construir el itinerario de aprendizaje. Para dotar de una mayor flexibilidad a la construcción del itinerario de aprendizaje se ha incluido en la ontología un concepto **Contents** que permite agrupar, gracias a la propiedad **has_contents**, aquellos elementos del dominio que pueden ser estudiados a la vez, es decir, no existe secuenciamiento entre ellos. De esta manera un punto en el itinerario de aprendizaje puede estar compuesto por un elemento individual o por un grupo de elementos.

El contexto del estudiante, es decir, los objetivos de aprendizaje alcanzados, se ha representado a través de las instancias de la ontología y el nivel de competencia alcanzado en cada una de ellas, representado como un valor real en el intervalo $[0, 10]$. Este nivel de competencia representa el nivel de conocimiento que un estudiante tiene sobre un elemento del dominio. A partir de los niveles de competencia alcanzados en estas instancias es posible calcular el nivel de competencia en las clases correspondientes. En concreto, calcularemos el nivel de competencia de una clase en un cierto lenguaje de

²OntoBridge es una librería para Java, desarrollada por el grupo GAIA de la Universidad Complutense de Madrid, con el fin de manejar ontologías de una forma sencilla. <http://gaia.fdi.ucm.es/research/ontobridge>

programación como la media aritmética de los niveles de competencia de sus instancias correspondientes a dicho lenguaje de programación.

5.2. Análisis del comportamiento y evaluación computacional de las estrategias propuestas

En esta sección presentamos un análisis del comportamiento y una evaluación experimental sobre la calidad de las estrategias de recomendación propuestas. El objetivo de esta evaluación es analizar las características que presentan las listas de objetos de aprendizaje recomendados y averiguar cuál es la mejor configuración de parámetros (tamaño de la lista de recomendación, nivel de personalización y nivel de diversidad) para las estrategias con el fin de generar unos prototipos que sean utilizados por usuarios reales.

En las siguientes secciones describiremos la metodología y los resultados obtenidos en la evaluación. En un primer momento haremos un análisis del comportamiento de las estrategias centrándonos en los niveles de *similitud*, *diversidad* y *utilidad pedagógica* que obtienen los conjuntos de objetos de aprendizaje recomendados para las distintas métricas de calidad propuestas en el Capítulo 3. Este análisis nos servirá para estudiar estas tres características de los conjuntos propuestos conforme varía el parámetro que ajusta las utilidades parciales en la métrica de calidad, pero sin tener en cuenta el tamaño de la lista de recomendación ni el orden en el que aparecen los objetos de aprendizaje. La similitud y la diversidad del conjunto de objetos de aprendizaje recomendados podríamos considerarlas ligadas a la satisfacción del estudiante con la recomendación y la utilidad pedagógica del conjunto de objetos de aprendizaje recomendados asociada con el impacto de las recomendaciones en el aprendizaje del estudiante. Posteriormente realizaremos una evaluación de la calidad de las listas de recomendación obtenidas con respecto a las tres características mencionadas –*similitud*, *diversidad* y *utilidad pedagógica*– teniendo en cuenta el tamaño de las listas y el orden de los objetos de aprendizaje en las mismas. Para ello realizaremos variaciones en los parámetros que afectan al comportamiento de ambas estrategias: α , que regula el nivel de personalización/diversidad introducido, y k , que establece el tamaño de la lista finalmente recomendada. Los resultados de esta última evaluación nos servirán para identificar cuál es el mejor valor para ambos parámetros que hacen que la calidad de las listas de recomendación sea alta.

Esta evaluación será realizada sobre la estrategia basada en conocimiento sensible al contexto que promueve altos niveles de personalización (en adelante, *KBpersonalization*) y sobre la estrategia basada en conocimiento que promueve la diversidad (en adelante, *KBdiversity*). La tercera estrategia, la estrategia de recomendación basada en conocimiento que combina proactividad y navegación por propuesta, como ya vimos en el Capítulo 3 (ver Sección 3.6), se diferencia de las anteriores en el modelo de interacción con

el usuario pero el proceso de recomendación puede seguir cualquiera de las dos aproximaciones anteriores, de ahí que el análisis de su comportamiento se encuentre cubierto con el análisis de las otras dos estrategias y ésta última estrategia sólo vaya a ser evaluada explícitamente desde el punto de vista de los usuarios.

5.2.1. Análisis del comportamiento general de las estrategias de recomendación

En esta primera sección presentamos un análisis del comportamiento de las estrategias *KBpersonalization* y *KBdiversity* en función de las relevancias parciales definidas (similitud, utilidad pedagógica y diversidad).

El análisis del comportamiento de las estrategias pretende mostrar cómo varían las relevancias parciales en la lista de recomendación a partir de variaciones del parámetro α en las métricas de calidad empleadas en la etapa de valoración, sin tener en cuenta el tamaño del conjunto de objetos de aprendizaje ni el orden de los mismos. Este comportamiento se analizará gracias a los valores medios de estas relevancias para el conjunto final de objetos de aprendizaje recomendados. En concreto, para cada estrategia y cada métrica de calidad sugerida en el Capítulo 3, compararemos los valores medios de dichas relevancias obtenidos por la estrategia de referencia que usa una aproximación de similitud pura –el parámetro α queda fijado a 1– con los valores medios de esas relevancias obtenidas con las versiones de la estrategia que emplean valores de α en el intervalo $[0, 1)$.

Así, para cada conjunto $P = \{L_1, L_2, \dots, L_n\}$ recomendado por una estrategia calcularemos:

- La similitud media \overline{Sim} de dicho conjunto como la media de las similitudes de cada objeto de aprendizaje con la consulta Q :

$$\overline{Sim}(P, Q) = \frac{\sum_{L_i \in P} Similitud(L_i, Q)}{|P|} \quad (5.1)$$

donde,

$$Similitud(L_i, Q) = \frac{|super(Q_conj_c) \cap super(L_i_conj_c)|}{\sqrt{|super(Q_conj_c)|} \cdot \sqrt{|super(L_i_conj_c)|}} \quad (5.2)$$

según vimos en el Capítulo 3.

- La utilidad pedagógica media \overline{UP} del conjunto P como la media de las utilidades pedagógicas de cada objeto de aprendizaje para el estudiante

S :

$$\overline{UP}(P, S) = \frac{\sum_{L_i \in P} UP(L_i, S)}{|P|} \quad (5.3)$$

donde,

$$UP(L_i, S) = 1 - MAN(L_i, S) \quad (5.4)$$

siendo $MAN(L_i, S)$ la media aritmética normalizada de los niveles de competencia alcanzados por el estudiante S en los conceptos cubiertos por L_i , según vimos en el Capítulo 3.

- La diversidad de dicho conjunto será medida a partir de la siguiente ecuación extraída de Smyth y McClave (2001)

$$Diversidad(P) = \frac{\sum_{i=1}^n \sum_{j=i}^n (1 - Similitud(L_i, L_j))}{\frac{n \cdot (n-1)}{2}} \quad (5.5)$$

donde la similitud entre dos objetos de aprendizaje L_i y L_j se calcula igual que entre un objeto de aprendizaje y la consulta (usando los conceptos que cubren).

Para la realización de estos experimentos hemos desarrollado 4 prototipos a partir del framework presentado en el Capítulo 4, dos relacionados con *KBpersonalization* y otros dos relacionados con *KBdiversity*.

Los prototipos de *KBpersonalization* tienen las siguientes características:

- La etapa de obtención de consulta se realiza a partir de un conjunto de consultas definidas en un fichero –clase `FromFileQueryElicitationStage`.
- La etapa de recuperación sigue una estrategia de recuperación aproximada –clase `ApproximateRetrievalStage`.
- La etapa de filtrado descarta aquellos objetos de aprendizaje que no están listos para ser explorados por el estudiante S a partir de la información contextual disponible –clase `ReachableConceptsFilter`.
- La etapa de valoración hace uso de una métrica de calidad compuesta por la agregación de dos utilidades parciales: la similitud con la consulta –clase `QuerySimilarity`– y la utilidad pedagógica –clase `PedagogicalUtility`.

- Uno de estos prototipos hará uso de la agregación que sigue el modelo de media ponderada –clase **WeightedMeanMetric**–, en adelante C_{1p} :

$$C_{1p}(L, S, Q) = \alpha \cdot \text{Similitud}(L, Q) + (1 - \alpha) \cdot UP(L, S) \quad \text{donde } \alpha \in [0, 1] \quad (5.6)$$

- El segundo de estos prototipos hará uso de la agregación que sigue el modelo de media armónica ponderada –la clase **HarmonicWeightedMeanMetric**–, en adelante C_{2p} :

$$C_{2p}(L, S, Q) = \frac{1}{\frac{\alpha}{\text{Similitud}(L, Q)} + \frac{(1-\alpha)}{UP(L, S)}} \quad \text{donde } \alpha \in [0, 1] \quad (5.7)$$

- La etapa de selección se queda con los k mejores elementos –clase **TopKSelectionStage**–, siendo k la mitad del tamaño del conjunto de objetos de aprendizaje que llegan a esta etapa.

Los prototipos desarrollados para *KBdiversity* en las etapas de obtención de consulta, recuperación y filtrado son similares a los anteriores. Para el resto de etapas tienen las siguientes características:

- La etapa de valoración hace uso de una métrica que estima la similitud con la consulta –clase **QuerySimilarity**.
- La etapa de selección por su parte hace uso de la estrategia de selección basada en diversidad para seleccionar k objetos de aprendizaje, siendo k la mitad del tamaño del conjunto de objetos de aprendizaje que llegan a esta etapa. Para calcular el valor de la calidad de un determinado objeto de aprendizaje a partir de la diversidad relativa:
 - Uno de estos prototipos hará uso de la agregación que sigue el modelo de media ponderada –clase **RelDWeightedMeanSelection**, en adelante C_{1d} :

$$C_{1d}(L, P, Q) = \alpha \cdot \text{Similitud}(L, Q) + (1 - \alpha) \cdot \text{RelD}(L, P) \quad \text{donde } \alpha \in [0, 1] \quad (5.8)$$

- El segundo de estos prototipos hará uso de la agregación que sigue el modelo de media armónica ponderada –clase **RelDHarmonicMeanSelection**, en adelante C_{2d} :

$$C_{2d}(L, P, Q) = \frac{1}{\frac{\alpha}{Similitud(L, Q)} + \frac{(1-\alpha)}{RelD(L, P)}} \quad \text{donde } \alpha \in [0, 1] \quad (5.9)$$

Donde P es el conjunto que se construye de manera incremental por el algoritmo de diversidad a partir del conjunto de objetos filtrados y valorados y el cómputo de $RelD(L, P)$ se hará conforme a la ecuación:

$$RelD(L, P) = \begin{cases} 1 & \text{si } P = \{\} \\ \frac{\sum_{L_i \in P} (1 - Similitud(L, L_i))}{|P|} & \text{en otro caso} \end{cases} \quad (5.10)$$

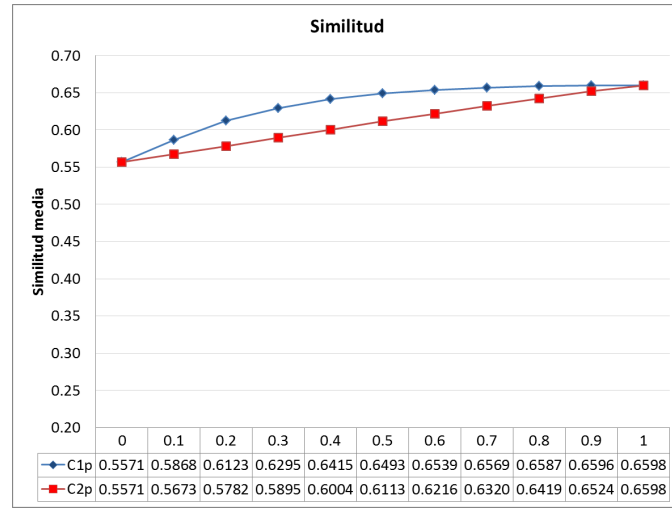
tal y como se indicó en el Capítulo 3.

Finalmente, hemos modelado un conjunto de 30 perfiles sintéticos y heterogéneos. Estos perfiles representan a estudiantes que han explorado aproximadamente el 80 % del camino de aprendizaje reflejado en la ontología y para todos los conceptos explorados tienen un nivel de competencia mayor o igual que 5 (5 es el valor del umbral de progreso en nuestros prototipos). Para cada perfil se han utilizado 18 consultas. De esta manera, se han realizado 540 diferentes recomendaciones para cada métrica de calidad. Asimismo hemos repetido este proceso de recomendación con diferentes valores de α en el intervalo 0 a 1 en intervalos de 0,1. Los valores de *similitud*, *utilidad pedagógica* y *diversidad* se han promediado para producir una única valoración para cada α .

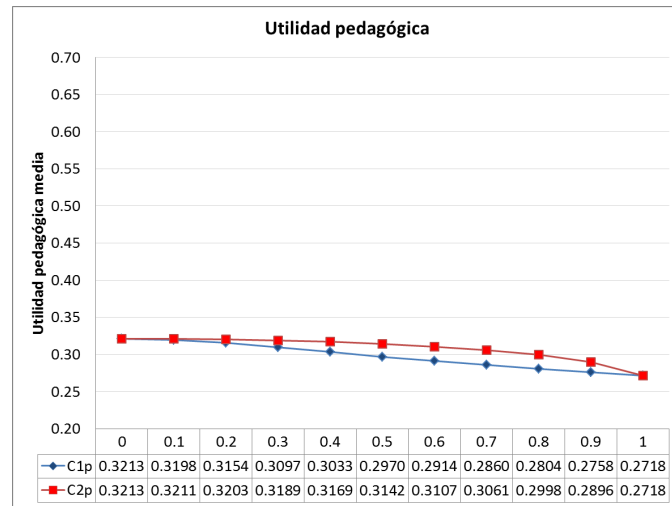
5.2.1.1. Resultados para la estrategia basada en conocimiento sensible al contexto que promueve altos niveles de personalización

En la Figura 5.2a podemos observar cómo se comportan ambas métricas de calidad, C_{1p} (Ecuación 5.6) y C_{2p} (Ecuación 5.7), con respecto a la similitud con la consulta. Para el caso base ($\alpha = 1$) el valor de similitud obtenido es 0,66. Como era de esperar este valor decrece conforme α se acerca al valor de máxima personalización ($\alpha = 0$) siendo el valor mínimo de similitud obtenido por ambas métricas de calidad 0,56. Si nos fijamos en la Figura 5.2a para C_{1p} se obtienen mejores valores de similitud, ya que la pérdida de la misma conforme α disminuye es menor que con C_{2p} .

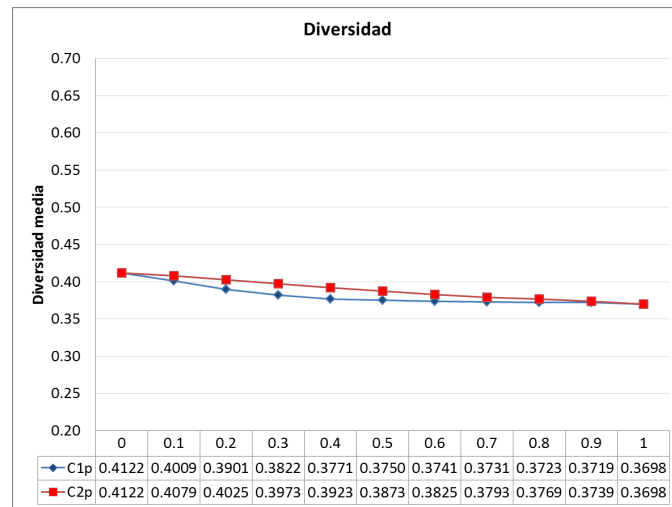
Desde el punto de vista de la utilidad pedagógica (Figura 5.2b), ambas métricas de calidad obtienen buenos resultados. Si nos fijamos en las gráficas vemos que el valor máximo de UP parece bajo (0,32). Sin embargo, no debemos olvidar que los perfiles diseñados se corresponden con alumnos que han explorado el 80 % del camino de aprendizaje y que tienen al menos



(a) Similitud



(b) Utilidad pedagógica



(c) Diversidad

Figura 5.2: Resumen de resultados sobre las relevancias parciales en la lista de objetos recomendados utilizando las métricas de calidad C_{1p} (Ecuación (5.6)) y C_{2p} (Ecuación (5.7)) en $KBpersonalization$

Tabla 5.1: Pérdida de similitud media con la consulta y ganancia de utilidad pedagógica media en base a la aproximación de referencia basada en similitud pura ($\alpha = 1$).

C_{1p}			C_{2p}		
α	Pérdida de \overline{Sim}	Ganancia de \overline{UP}	α	Pérdida de \overline{Sim}	Ganancia de \overline{UP}
0,0	15,56 %	18,22 %	0,0	15,56 %	18,22 %
0,1	11,06 %	17,65 %	0,1	14,02 %	17,95 %
0,2	7,19 %	16,03 %	0,2	12,37 %	17,67 %
0,3	4,58 %	13,94 %	0,3	10,65 %	17,15 %
0,4	2,76 %	11,58 %	0,4	9,00 %	16,41 %
0,5	1,59 %	9,28 %	0,5	7,34 %	15,42 %
0,6	0,88 %	7,21 %	0,6	5,78 %	14,13 %
0,7	0,44 %	5,22 %	0,7	4,21 %	12,45 %
0,8	0,16 %	3,15 %	0,8	2,71 %	10,14 %
0,9	0,03 %	1,46 %	0,9	1,12 %	6,39 %
1,0	0,00 %	0,00 %	1,0	0,00 %	0,00 %

un nivel de competencia de 5 sobre 10 en todos los conceptos explorados. Esto supone que la máxima ganancia de UP que puede conseguirse para el 80 % de los conceptos es 0,5. Si nos fijamos en los resultados, cuando $\alpha = 1$ (no se tiene en cuenta la información contextual en la etapa de valoración) observamos que la UP media obtenida supera el valor 0,27 lo que podemos considerar un buen valor, teniendo en cuenta que la información contextual sólo se ha tenido en cuenta en la etapa de filtrado.

La Tabla 5.1 muestra la pérdida de similitud media con la consulta y la ganancia de utilidad pedagógica media con respecto a la estrategia de referencia (similitud pura, $\alpha = 1$) obtenida por cada métrica de calidad en función de α . Vemos que ambas métricas obtienen, para el mejor caso ($\alpha=0$), más de un 18 % de ganancia de \overline{UP} con respecto a la estrategia de referencia (que no tiene en cuenta la utilidad pedagógica en la etapa de valoración) a costa de perder en torno al 15,5 % de similitud con la consulta. La Tabla 5.1 muestra que la estrategia de recomendación propuesta alcanza un compromiso entre la ganancia de utilidad pedagógica y la pérdida de similitud con ambas métricas, siendo la pérdida de similitud menor que la ganancia de utilidad pedagógica. Consideramos que éste es un comportamiento satisfactorio de acuerdo a nuestro principal objetivo, obtener recomendaciones que cumplan la consulta del estudiante mientras que se adaptan a la información contextual de la actividad y del propio estudiante.

Aunque esta estrategia no se centra en promover la diversidad del conjunto final queremos ver cuál es la media de diversidad obtenida para ambas métricas de calidad en esta estrategia. Estos resultados están reflejados en

la Figura 5.2c. Podemos observar que cuando el valor de similitud es máximo ($\alpha = 1$) la diversidad media obtenida está próxima a 0,37. En el caso contrario, cuando la similitud no se tiene en cuenta para valorar un objeto de aprendizaje ($\alpha = 0$), la diversidad alcanza su valor máximo, siendo éste cercano a 0,41.

5.2.1.2. Resultados para la estrategia basada en conocimiento que promueve la diversidad

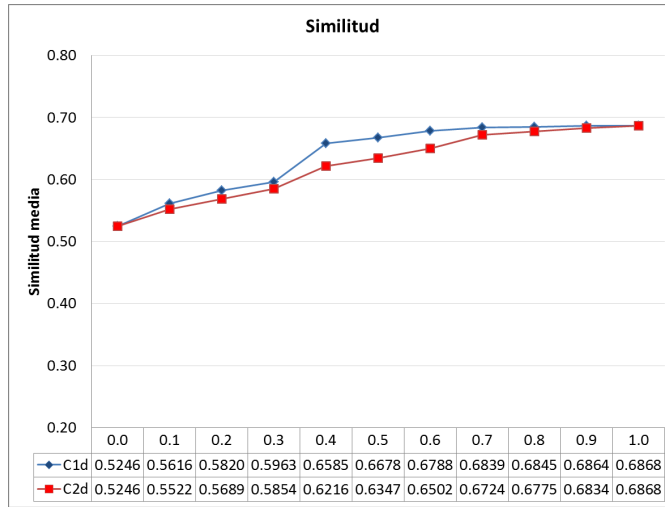
La Figura 5.3 resume los resultados obtenidos para los valores medios de las propiedades propuestas en el prototipo *KBdiversity* con ambas métricas de calidad y en función del parámetro α .

En la Figura 5.3 también podemos ver, como era de esperar, que la similitud aumenta con α (Figura 5.3a) y la diversidad, por el contrario, cae a medida que el parámetro α aumenta (Figura 5.3c). Si comparamos ambos prototipos, los valores de similitud media no difieren de una manera significativa en los valores límites de α . Esto mismo sucede con los valores de diversidad. Sin embargo, para valores más centrados en el intervalo $[0,1]$ observamos comportamientos ligeramente diferentes en ambos prototipos: la similitud media es mayor para C_{1d} (Ecuación (5.8)) que para C_{2d} (Ecuación (5.9)), mientras que C_{2d} obtiene mayores valores de diversidad que C_{1d} .

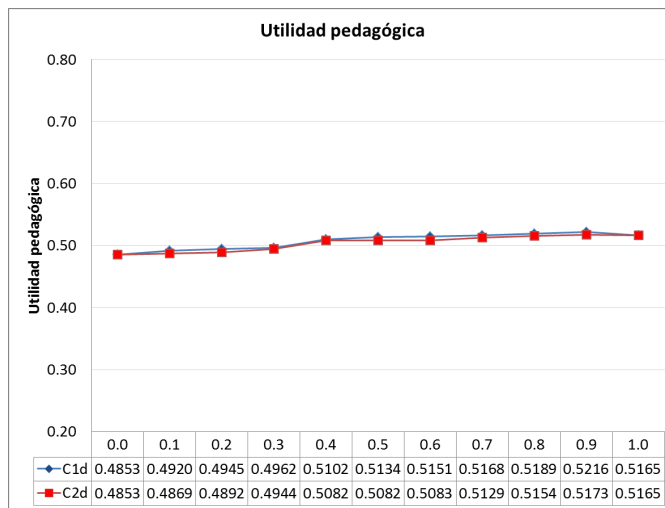
Desde el punto de vista de la utilidad pedagógica ambas métricas de calidad obtienen valores similares de utilidad pedagógica media, que van desde 0,48 hasta cerca de 0,52 (ver Figura 5.3b). La inclusión de la diversidad reduce ligeramente la utilidad pedagógica media en comparación con la aproximación basada en similitud pura ($\alpha = 1$). Sin embargo, estos resultados de *UP* nos muestran que *KBdiversity* obtiene altos niveles de *UP* para un estudiante aún cuando este prototipo sólo hace uso de la información contextual en la etapa de filtrado. Al igual que sucedía con el prototipo anterior, de acuerdo a los perfiles utilizados la ganancia máxima de *UP* para el 80 % de los conceptos sólo puede ser 0,5. Estos resultados en la *UP* por el prototipo *KBdiversity* nos hacen ver que esta estrategia de manera colateral consigue la inclusión de conceptos de ese 20 % de conceptos para los que la *UP* es mayor.

La Tabla 5.2 muestra la pérdida de similitud media con la consulta así como la ganancia de diversidad media entre los objetos de aprendizaje recomendados en relación con la estrategia de referencia (similitud pura, $\alpha = 1$) obtenida por cada métrica de calidad para distintos valores de α .

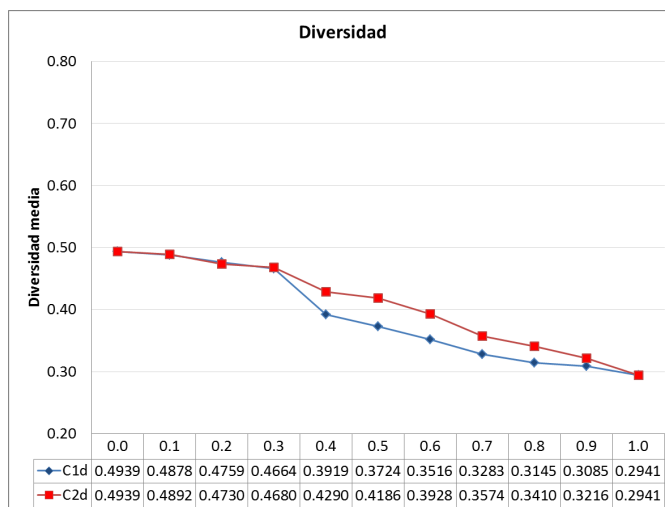
Si nos fijamos en la pérdida de similitud media, los resultados muestran que en el peor de los casos (valor mínimo de similitud media) las recomendaciones generadas pierden un 23,62 % de similitud con la consulta. Estos resultados recalcan que incluso cuando la similitud no es tenida en cuenta para ordenar los objetos de aprendizaje recomendados ($\alpha = 0$) la estrategia de recomendación propuesta obtiene altos valores de similitud, por lo tanto



(a) Similitud



(b) Utilidad pedagógica



(c) Diversidad

Figura 5.3: Resumen de resultados sobre las relevancias parciales en la lista de objetos recomendados utilizando las métricas de calidad C_{1d} (Ecuación (5.8)) y C_{2d} (Ecuación (5.9)) en $KBdiversity$

Tabla 5.2: Pérdida de similitud media con la consulta y ganancia de diversidad en base a la aproximación de referencia basada en similitud pura ($\alpha = 1$).

C_{1d}			C_{2d}		
α	Pérdida de \overline{Sim}	Ganancia de $Diversidad$	α	Pérdida de \overline{Sim}	Ganancia de $Diversidad$
0,0	23,62 %	40,46 %	0,0	23,62 %	40,46 %
0,1	18,23 %	39,23 %	0,1	19,35 %	38,56 %
0,2	15,26 %	36,81 %	0,2	16,92 %	35,28 %
0,3	13,18 %	34,88 %	0,3	14,51 %	34,27 %
0,4	4,12 %	19,80 %	0,4	9,22 %	26,36 %
0,5	2,76 %	15,86 %	0,5	7,31 %	24,26 %
0,6	1,17 %	11,65 %	0,6	5,04 %	19,05 %
0,7	0,42 %	6,94 %	0,7	1,81 %	11,87 %
0,8	0,33 %	4,14 %	0,8	1,06 %	8,55 %
0,9	0,06 %	2,92 %	0,9	0,19 %	4,62 %
1,0	0,00 %	0,00 %	1,0	0,00 %	0,00 %

los objetivos a corto plazo del estudiante se conservan.

Desde el punto de vista de la diversidad, ambos prototipos obtienen, para el caso mejor ($\alpha = 0$), más de un 40 % de ganancia de diversidad con respecto al caso base –similitud pura.

La Tabla 5.2 muestra que la estrategia de recomendación propuesta con una etapa de selección basada en diversidad obtiene un compromiso en ambos prototipos entre la ganancia de diversidad y la pérdida de similitud, siendo la pérdida de similitud mucho menor que la ganancia de diversidad. A la vista de estos resultados consideramos que este comportamiento mostrado por la estrategia es satisfactorio de acuerdo a nuestro principal objetivo, mejorar la diversidad del conjunto de objetos de aprendizaje recomendados sin penalizar significativamente la similitud con la consulta.

5.2.2. Evaluación experimental de la calidad de las listas de recomendación considerando su tamaño y el orden de sus elementos

En esta sección realizaremos una evaluación sobre la calidad de las listas de recomendación proporcionadas por las estrategias *KBpersonalization* y *KBdiversity* teniendo en cuenta el orden y el tamaño de dichas listas. Esta calidad se evaluará de acuerdo a tres relevancias: la similitud con la consulta, y la utilidad pedagógica (en el caso de *KBpersonalization*) o la diversidad (en el caso de *KBdiversity*). En particular, analizaremos el impacto que tiene el

parámetro α en las métricas de calidad propuestas (las respectivas C_1 y C_2 en cada estrategia) y el impacto del tamaño de la lista de recomendación (Ruiz-Iniesta et al., 2011b), representado por el parámetro k .

Debido a nuestro interés por tener listas de recomendación pequeñas con el fin de evitar la sobrecarga de recursos al estudiante, consideramos crucial que los objetos de aprendizaje más relevantes aparezcan en las primeras posiciones de la lista de recomendación. Existen varias métricas en el ámbito de la Recuperación de Información que tienen en cuenta el orden de los documentos en una lista de resultados. En particular, la métrica conocida como *Normalized Discounted Cumulative Gain (NDCG)* tiene una gran acogida (Järvelin y Kekäläinen, 2002). Esta métrica mide la utilidad o ganancia de una lista de resultados basada en la relevancia y la posición de los documentos recuperados y se compara la ganancia obtenida con la ideal. Esta métrica será la que emplearemos para analizar el comportamiento de la estrategia con distintos tamaños en la lista de recomendación.

Para nuestro propósito, hemos adaptado la métrica *NDCG* con el fin de analizar la lista de objetos de aprendizaje recomendados $L = \{L_1, L_2, \dots, L_k\}$, para un estudiante S y en respuesta a una consulta Q , respecto a su similitud con la consulta ($NDCG_{Sim}$, Ecuación 5.11) y su utilidad para el estudiante ($NDCG_{UP}$, Ecuación 5.12) o a la diversidad de los objetos de aprendizaje ($NDCG_{Div}$, Ecuación 5.13) según corresponda en atención a la estrategia que se esté estudiando (*KBPersonalization* o *KBdiversity*). Calcularemos estos valores para listas de tamaño k utilizando las siguientes ecuaciones:

$$\begin{aligned} NDCG_{Sim}(L, Q) &= \frac{DCG_{Sim}(L, Q)}{IDCG_{Sim}(L', Q)} \\ &= \frac{Similitud(L_1, Q) + \sum_{i=2}^k \frac{Similitud(L_i, Q)}{\log_2 i}}{IDCG_{Sim}(L', Q)} \end{aligned} \quad (5.11)$$

$$\begin{aligned} NDCG_{UP}(L, S) &= \frac{DCG_{UP}(L, S)}{IDCG_{UP}(L', S)} \\ &= \frac{UP(L_1, S) + \sum_{i=2}^k \frac{UP(L_i, S)}{\log_2 i}}{IDCG_{UP}(L', S)} \end{aligned} \quad (5.12)$$

$$\begin{aligned} NDCG_{Div}(L) &= \frac{DCG_{Div}(L)}{IDCG_{Div}(L')} \\ &= \frac{RelD(L_1, \{\}) + \sum_{i=2}^k \frac{RelD(L_i, \{L_1, L_2, \dots, L_{i-1}\})}{\log_2 i}}{IDCG_{Div}(L')} \end{aligned} \quad (5.13)$$

Donde $IDCG_{UP}(L', S)$ e $IDCG_{Sim}(L', Q)$ son los valores *DCG* de la lista L' resultado de ordenar L por la utilidad pedagógica de los objetos

de aprendizaje para el estudiante S y por la similitud de cada objeto de aprendizaje con respecto a la consulta Q , respectivamente. En el caso de $NDCG_{Div}$, el valor de $DCG_{Div}(L)$ se obtiene a partir de la diversidad relativa calculada para cada objeto de aprendizaje según la Ecuación 5.10 en la etapa de selección de la estrategia. Por su parte el valor de $IDCG_{Div}(L')$ es el valor de DCG de la lista L' que se obtiene reordenando los elementos de L aplicando el algoritmo de selección por diversidad de Smyth y McClave (2001), donde la métrica de calidad utilizada será únicamente la diversidad relativa. La lista L' , del mismo tamaño que L , tiene como primer elemento aquel con mayor valor de similitud con la consulta (calculado previamente en la etapa de valoración) y los siguientes serán introducidos según la diversidad relativa mostrada con la lista que se va construyendo en cada paso del algoritmo.

Para concluir esta evaluación, también queremos averiguar cuál es el mejor valor de α para ambas estrategias, es decir, el valor de α que garantiza el más alto compromiso entre la personalización/diversidad y la similitud. En otras palabras, el mejor valor de α es aquel para el que la pérdida de utilidad pedagógica/diversidad relativa al valor óptimo de utilidad/diversidad ($\alpha = 0$) y la pérdida de similitud relativa al valor óptimo de similitud ($\alpha = 1$) es mínimo. Para este propósito, hemos definido funciones que representan este compromiso:

$$\begin{aligned} Comp_{\alpha,UP-Sim}(L, S, Q) = & \\ & (NDCG_{UP}(L, S)_{\alpha=0} - NDCG_{UP}(L, S)_{\alpha}) + \\ & + (NDCG_{Sim}(L, Q)_{\alpha=1} - NDCG_{Sim}(L, Q)_{\alpha}) \end{aligned} \quad (5.14)$$

$$\begin{aligned} Comp_{\alpha,Div-Sim}(L, Q) = & \\ & (NDCG_{Div}(L)_{\alpha=0} - NDCG_{Div}(L)_{\alpha}) + \\ & + (NDCG_{Sim}(L, Q)_{\alpha=1} - NDCG_{Sim}(L, Q)_{\alpha}) \end{aligned} \quad (5.15)$$

Los prototipos empleados en esta segunda evaluación son los mismos que detallamos en la sección anterior (ver Sección 5.2.1).

Finalmente, hemos utilizado el mismo conjunto de perfiles y consultas que en el caso anterior, 30 perfiles sintéticos y heterogéneos que han explorado aproximadamente el 80 % del camino de aprendizaje y 18 consultas para cada perfil. De nuevo hemos obtenido 540 diferentes recomendaciones para cada métrica de calidad. Asimismo hemos repetido este proceso de recomendación con diferentes valores de α –en el intervalo 0 a 1 en intervalos de 0,1– y diferentes tamaños para la lista de recomendados – $k = 5, 10$ y 20 . Tanto los valores de *Similitud*, *UP*, *Diversidad* y el valor *NDCG* sobre todas las listas

recomendadas se han promediado para producir una única valoración para cada par de valores de α y k .

5.2.2.1. Resultados para la estrategia basada en conocimiento sensible al contexto que promueve altos niveles de personalización

Para facilitar el análisis y la lectura de los resultados de esta evaluación hemos separado los mismos en función de las métricas de calidad. La Figura 5.4 resume los resultados para la métrica de calidad C_{1p} (Ecuación 5.6), utilizando las métricas de evaluación propuestas, agrupados por el tamaño de las listas recomendadas y trazados conforme a α . Podemos observar en los resultados que la tendencia de $NDCG_{Sim}$ (Figura 5.4a) no difiere de manera muy significativa dependiendo de α ni del tamaño de la lista (para un valor de α fijado). La gráfica muestra que la estrategia de recomendación obtiene valores altos de ganancia de similitud. A pesar de utilizar un método de recuperación aproximada, los resultados para $NDCG_{Sim}$ muestran que la similitud con la consulta se preserva en la lista de objetos recomendados. Consideramos que éste es un buen comportamiento. Con estos resultados podemos ver que *KBpersonalization* siempre garantiza que los objetos de aprendizaje propuestos satisfacen los objetivos a corto plazo del estudiante (representados en la consulta propuesta). Incluso con el mayor nivel de personalización ($\alpha=0$), donde la consulta se utiliza sólo en la etapa de recuperación y la métrica de calidad se expresa en términos de la utilidad pedagógica, los resultados garantizan un alto valor de similitud.

En cuanto a la tendencia de los valores para $NDCG_{UP}$ (Figura 5.4b), se diferencian en función de α y del tamaño de la lista (especialmente para valores altos de α). Sin embargo, cualquiera de los resultados obtenidos se mantiene en valores altos para $NDCG_{UP}$ (el valor más bajo obtenido, para $k=5$ y $\alpha=1,0$ es 0,790).

KBpersonalization alcanza valores altos en ambas medidas incluso cuando el tamaño de la lista de recomendación es pequeña ($k=5$). Además, a medida que k crece, el valor $NDCG_x$ del término x (*Similitud* o *UP*) que está penalizado por el valor de α en la Ecuación (5.6) también aumenta. Por ejemplo, un valor bajo de α penaliza los intereses a corto plazo (*Similitud*) en la Ecuación (5.6). Sin embargo, a medida que k crece, los objetos de aprendizaje más similares con la consulta que no fueron incluidos en la lista de recomendación tienen más probabilidad de conseguir ser ahora incluidos y consecuentemente el valor $NDCG_{Sim}$ aumenta.

La Figura 5.5 representa los resultados del cálculo del compromiso entre la similitud y la utilidad pedagógica usando la métrica de calidad C_{1p} . Se puede observar que los valores de α en el intervalo $[0,1, 0,3]$ obtienen la pérdida mínima. $\alpha = 0,2$ permite lograr el mejor compromiso entre los objetivos a corto y largo plazo y puede ser considerado como el valor ideal de α para la

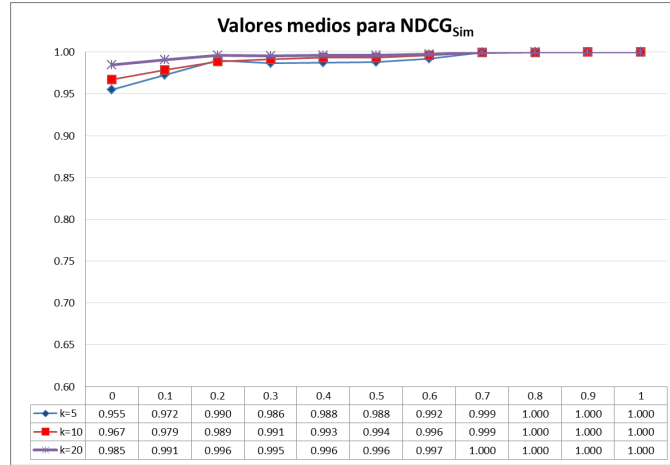
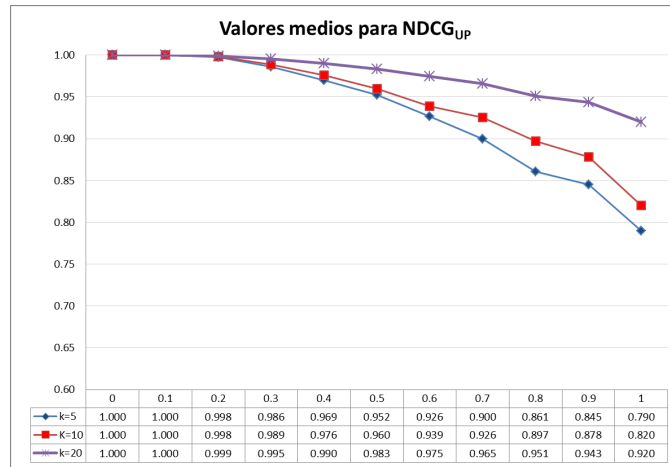
(a) $NDCG_{Sim}$ (b) $NDCG_{UP}$

Figura 5.4: Comparación de $NDCG_{Sim}$ y $NDCG_{UP}$ en función de α y k en $KBpersonalization$ con C_{1p} (Ecuación 5.6) como métrica de calidad

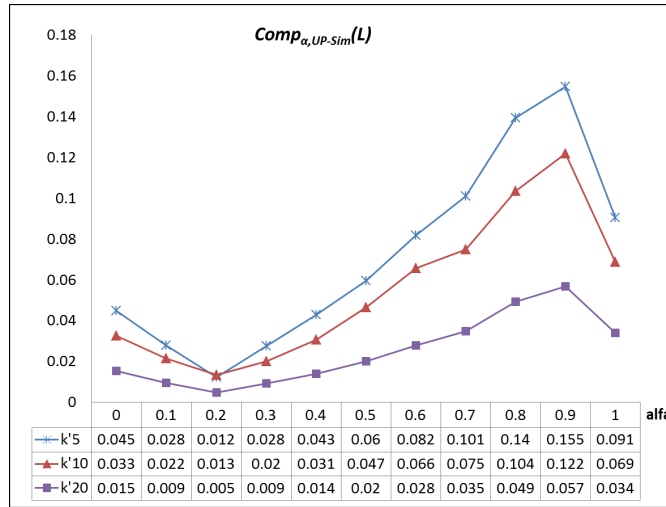


Figura 5.5: Función de compromiso para *KBpersonalization* con C_{1p} (Ecuación 5.6).

configuración particular de la estrategia genérica utilizada en el experimento.

En cuanto a los resultados utilizando C_{2p} (Ecuación 5.7), la Figura 5.6 los resume utilizando las métricas de evaluación propuestas, agrupados por el tamaño de la lista recomendada y trazados conforme a α .

La tendencia de $NDCG_{Sim}$ (Figura 5.6a), a diferencia del caso anterior, presenta una mayor diferencia para los distintos valores de α , sobre todo para valores pequeños del tamaño de la lista de recomendación. Podemos observar que a menor tamaño de la lista, la similitud con la consulta de los objetos de aprendizaje para valores pequeños de α es menor. Esto es debido al método de recuperación utilizado (recuperación aproximada), y a que la métrica de calidad quiere compensar la utilidad pedagógica de los objetos de aprendizaje a costa de penalizar la similitud. Sin embargo, observamos que en el caso peor ($\alpha=0$) la ganancia de similitud es 0,766, que sigue siendo un valor alto.

En cuanto a los resultados para $NDCG_{UP}$ observamos que la ganancia de UP difiere para valores de k a medida que el valor de α aumenta (Figura 5.6b). Podemos observar que para valores de α mayores que 0,8 la ganancia de utilidad pedagógica se ve más penalizada. Sin embargo el mínimo valor obtenido es 0,748 (para $\alpha=1$ y $k=5$) un valor relativamente alto y aceptable de ganancia de utilidad pedagógica.

Al igual que sucedía con la anterior métrica de calidad, la estrategia de recomendación alcanza valores altos en ambas medidas incluso cuando el tamaño de la lista de recomendación es pequeña ($k=5$) y debería ser más difícil cumplir ambos objetivos.

La Figura 5.7 representa el compromiso de *KBpersonalization* con la

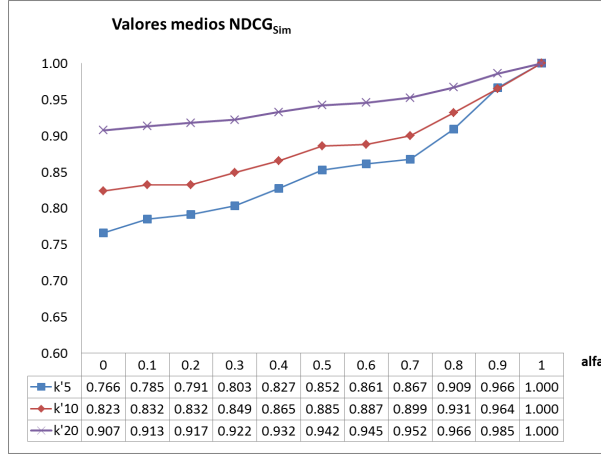
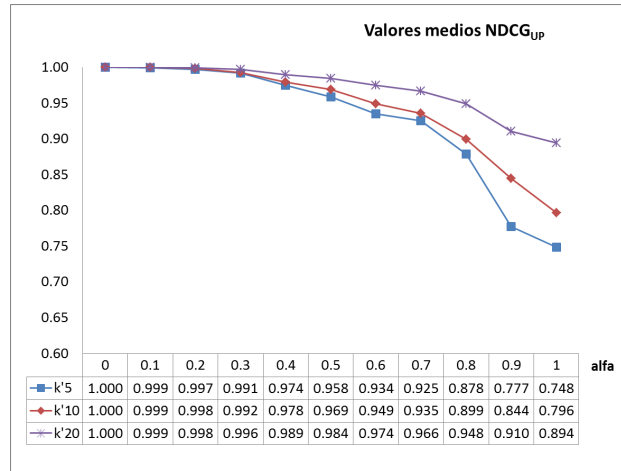
(a) $NDCG_{Sim}$ (b) $NDCG_{UP}$

Figura 5.6: Comparación de $NDCG_{Sim}$ y $NDCG_{UP}$ en función α y k en $KBpersonalization$ con C_{2p} (Ecuación 5.7) como métrica de calidad.

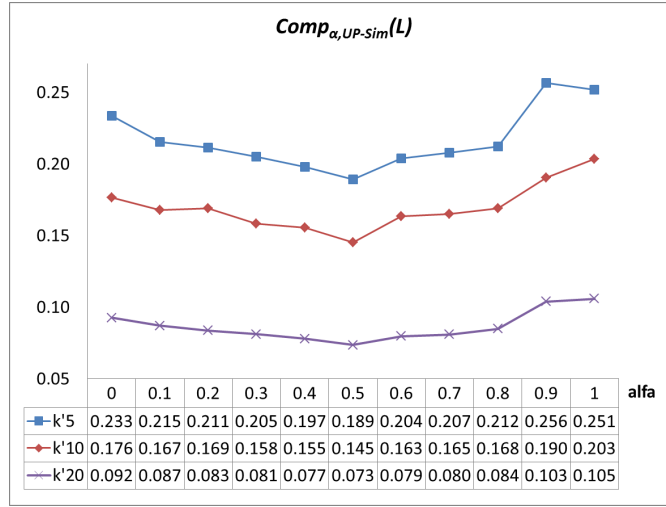


Figura 5.7: Función de compromiso para *KBpersonalization* con C_{2p} (Ecuación 5.7).

métrica C_{2p} , donde se puede observar que los valores de α en el intervalo $[0, 4, 0, 6]$ obtienen la pérdida mínima. $\alpha = 0,5$ permite lograr el mejor compromiso entre los objetivos a corto y largo plazo y puede ser considerado como el valor ideal de α para la configuración particular de la estrategia genérica utilizada en el experimento.

Aunque los resultados obtenidos para ambas métricas de calidad son prometedores (tanto en similitud como en utilidad pedagógica las ganancias respectivas son altas), podemos observar que la métrica de calidad C_{1p} obtiene mejores resultados. Si nos fijamos en la función de compromiso, para la métrica de C_{1p} los valores de esta función son menores que los respectivos para la función compromiso de la métrica de C_{2p} . Esto es algo que ya se intuía en la Tabla 5.1 para un estudio de grano más grueso donde no se tenía en cuenta ni el número de elementos ni el orden entre ellos. De cara a la experiencia con usuarios reales el prototipo de la estrategia *KBpersonalization* que se usará será el que utiliza la métrica C_{1p} .

5.2.2.2. Resultados para la estrategia basada en conocimiento que promueve la diversidad

La Figura 5.8 resume los resultados para la métrica de calidad C_{1d} de la Ecuación 5.8, utilizando las métricas de evaluación propuestas, agrupados por el tamaño de las listas recomendadas y trazados conforme a α .

Podemos observar en los resultados que $NDCG_{Sim}$ (Figura 5.8a), dependiendo de α y del tamaño de la lista, aumenta conforme aumenta el valor de α y además obtiene altos valores de ganancia de similitud. Resulta quizá llamativo el salto que da la ganancia de similitud para valores de

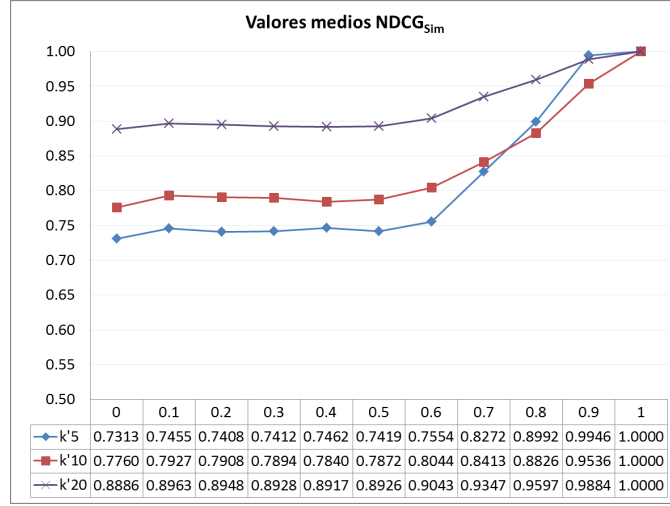
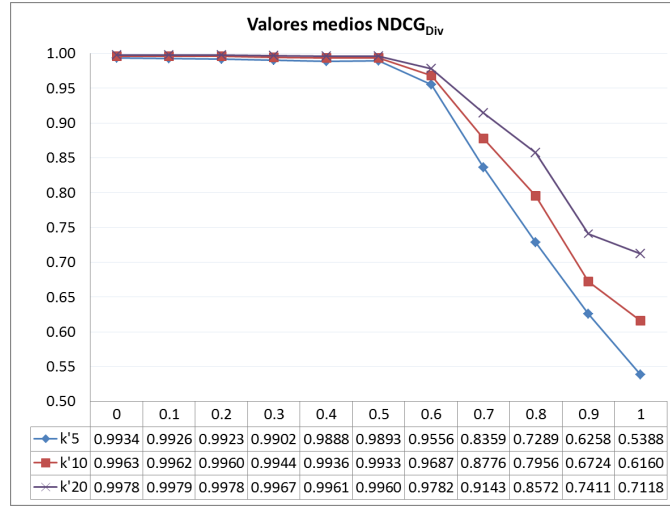
(a) $NDCG_{Sim}$ (b) $NDCG_{Div}$

Figura 5.8: Comparación de $NDCG_{Sim}$ y $NDCG_{Div}$ en función de α y k en $KBdiversity$ con C_{1d} (Ecuación (5.8)) como métrica de calidad.

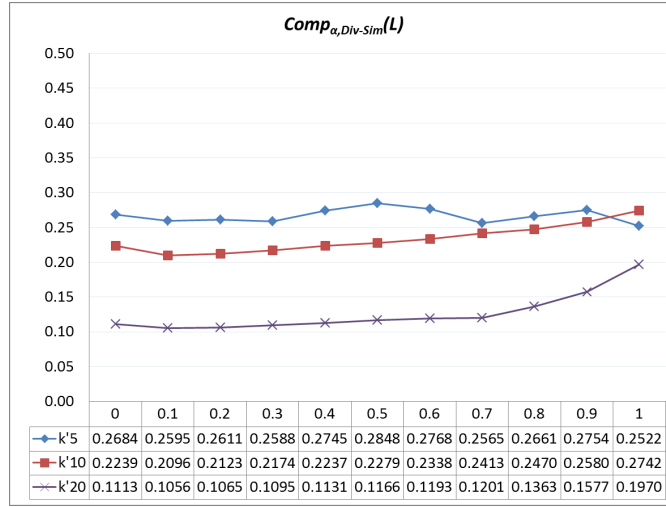


Figura 5.9: Función de compromiso para $KBdiversity$ con C_{1d} (Ecuación 5.8).

$\alpha > 0,6$, más aún para valores pequeños de k . Más adelante observaremos este mismo comportamiento con respecto a la diversidad. Aún así y a pesar de utilizar un método de recuperación aproximada, los resultados para $NDCG_{Sim}$ muestran que la similitud con la consulta se preserva en la lista de objetos recomendados. Este comportamiento garantiza que los objetos de aprendizaje propuestos satisfacen los objetivos a corto plazo del estudiante (representados en la consulta propuesta), incluso con el mayor nivel de diversidad ($\alpha = 0$), donde la consulta (objetivos a corto plazo) se utiliza sólo en la etapa de recuperación y la métrica de calidad se expresa en términos de la diversidad relativa.

En cuanto a la tendencia de los valores para $NDCG_{Div}$ (Figura 5.8b), cuando el valor de $\alpha < 0,6$ no se diferencian significativamente en función de α y del tamaño de la lista. Además observamos que para esos valores de α se obtienen grandes ganancias de diversidad. En cambio, cuando $\alpha > 0,6$ la caída de la diversidad se ve disparada. Consideramos que estos resultados son favorables, ya que en el caso peor (no existe diversidad en la etapa de ordenación, $\alpha = 1$) la ganancia de diversidad está cercana a 0,62 para $k = 10$ a pesar de obtener una ganancia de similitud, en esos mismos valores de k y α , igual a 1,0. En general, podemos insistir en que $KBdiversity$ obtiene valores altos de diversidad, tal y como apuntábamos en los resultados del análisis del comportamiento.

En la Figura 5.9 tenemos representados los resultados para el compromiso entre la similitud y la diversidad usando C_{1d} . En este caso podemos observar que apenas hay diferencia entre valores de α menores que 0,5, aunque podemos decir que $\alpha = 0,1$ permite lograr el mejor compromiso entre la similitud y la diversidad.

Por último, pasemos a analizar los resultados para C_{2d} . La Figura 5.10 resume los resultados para C_{2d} (Ecuación (5.9)) agrupados por el tamaño de la lista recomendada y trazados conforme a α .

La tendencia de $NDCG_{Sim}$ (Figura 5.10a) es muy similar al caso anterior. En este segundo prototipo podemos observar que la ganancia de similitud aumenta de manera más progresiva desde los valores más bajos de α hasta alcanzar el máximo nivel de similitud en $\alpha = 1$.

En cuanto a los resultados para $NDCG_{Div}$ observamos que la ganancia de *Diversidad* a medida que el valor de α aumenta, disminuye, como era de esperar (Figura 5.10b). En general, podemos decir que no existe una diferencia apreciable para los distintos tamaños de la lista de recomendación para valores de $\alpha < 0,5$, pero cuando $\alpha > 0,5$ esta diferencia se ve incrementada. Finalmente también observamos que esta métrica de calidad permite obtener mayores valores de diversidad que la métrica anterior (C_{1d}). Con la métrica de calidad C_{2d} observamos que cuando $\alpha < 0,3$ la ganancia de diversidad es prácticamente igual en los tres valores de α y k . A medida que α aumenta la ganancia de diversidad va cayendo hasta llegar a sus valores mínimos cuando $\alpha = 1$, que para $k = 10$ es 0,722.

Para terminar, el compromiso de esta métrica de calidad está reflejado en la Figura 5.11. En esta ocasión existen diferencias poco significativas para valores de α menores que 0,6. Aún así, podemos decir que los valores de α en el intervalo $[0, 0,2]$ obtienen la pérdida mínima. Al igual que con la anterior métrica de calidad $\alpha = 0,1$ permite lograr el mejor compromiso entre la similitud y la diversidad y puede ser considerado como el valor ideal de α para la configuración particular de la estrategia.

Los resultados obtenidos con ambas métricas de calidad son prometedores. Para $\alpha = 0,1$ ambas métricas permiten lograr el mejor compromiso entre la similitud y la diversidad, ofreciendo mejores resultados C_{2d} . De ahí que en la experiencia con usuarios reales el prototipo de la estrategia *KBdiversity* que se usará será el que emplea la métrica C_{2d} .

5.3. Evaluando las estrategias de recomendación con profesores y alumnos

Después de haber realizado el análisis experimental de las estrategias, y gracias a la determinación de los valores de α mejores para cada métrica de calidad, desarrollamos un conjunto de prototipos funcionales para así evaluar en un entorno real (con estudiantes y profesores) el nivel de aceptación de las estrategias.

El objetivo principal que persigue esta evaluación es determinar la acogida que tienen las herramientas por parte de los potenciales usuarios de las mismas (profesores y estudiantes) como una nueva manera de acceder a los

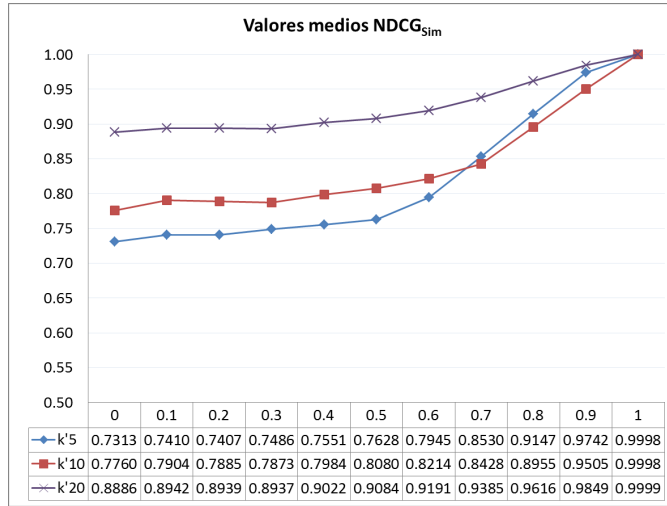
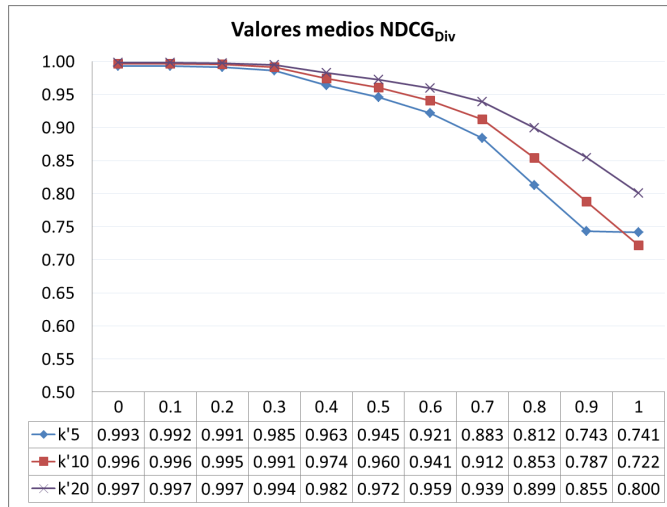
(a) $NDCG_{Sim}$ (b) $NDCG_{Div}$

Figura 5.10: Comparación de $NDCG_{Sim}$ y $NDCG_{Div}$ con α y k en *KBdiversity* con C_{2d} (Ecuación (5.9)) como métrica de calidad.

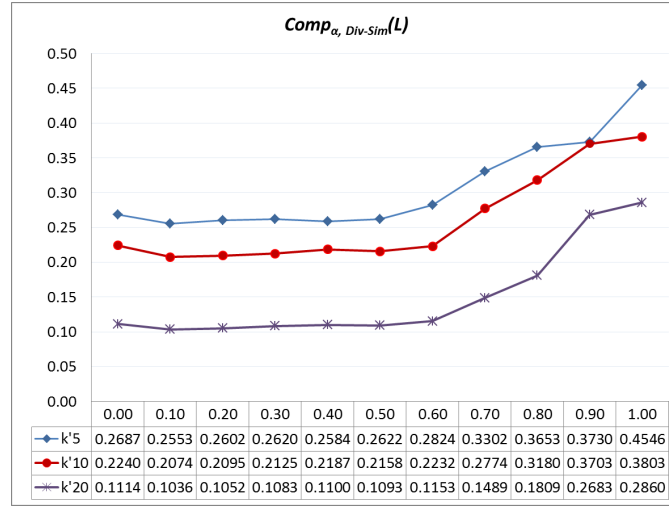


Figura 5.11: Función de compromiso para $KBdiversity$ con C_{2d} (Ecuación (5.9)).

repositorios de objetos de aprendizaje. De esta manera, podremos obtener una valoración sobre el grado de aceptación y su utilidad esperable para el apoyo a la enseñanza. Esta evaluación nos permitirá además obtener una retroalimentación sobre mejoras que se puedan realizar en las estrategias de recomendación.

Para la realización de esta evaluación hemos decidido emplear el método de evaluación guiado por objetivos (GQM) propuesto por Basili y Rombach (1988). Este método señala que con el fin de mejorar el proceso de evaluación debemos definir objetivos medibles, que deben ser refinados en cuestiones, y, consecuentemente, en métricas que nos darán toda la información para contestar dichas cuestiones. El análisis y la interpretación de las respuestas nos ayudará a conocer cuáles de los objetivos propuestos han sido cumplidos (Solingen y Berghout, 1999).

Nuestro objetivo principal puede ser dividido en tres sub-objetivos basados en los propuestos por la teoría unificada de aceptación de los usuarios de las tecnologías de la información (Venkatesh et al., 2003):

1. Valoración de rendimiento, el grado en el que un individuo cree que la herramienta de recomendación le ayudará a conseguir sus objetivos de aprendizaje.
2. Valoración de esfuerzo, el grado de facilidad de uso de la herramienta.
3. Valoración de intención de uso futuro, el grado de utilización que tendría la herramienta si estuviera disponible en otros dominios de aprendizaje.

Tabla 5.3: Preguntas relacionadas a cada uno de los objetivos de la evaluación

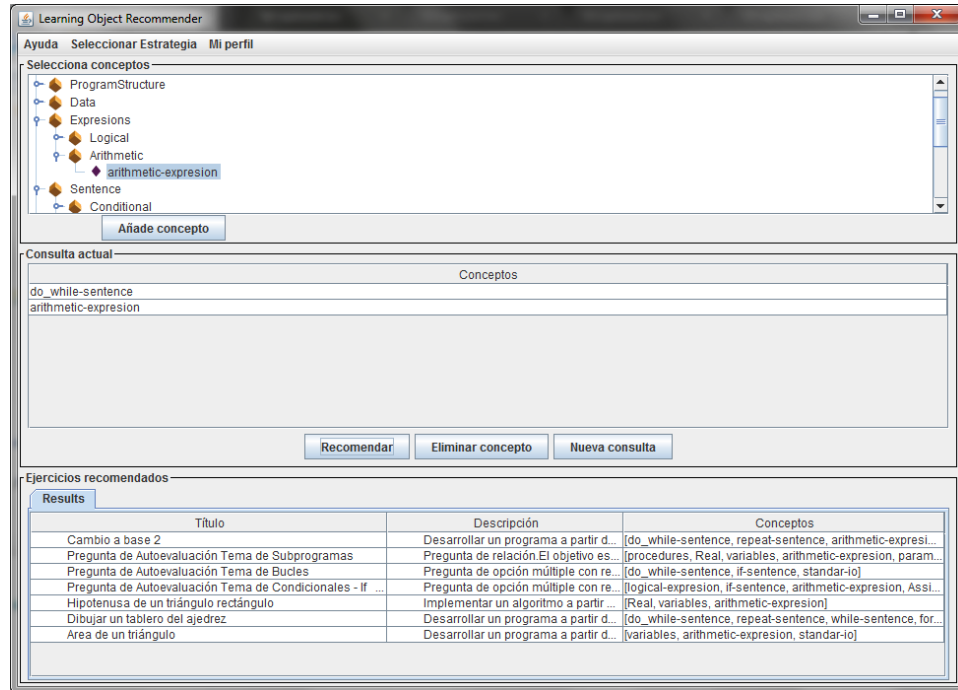
Rendimiento	
P1	Creo que es positivo para el aprendizaje disponer de una herramienta de recomendación con ejercicios resueltos, ejemplos y preguntas.
P2 ¹	La herramienta de recomendación proporciona ejercicios relevantes para practicar la asignatura adaptados al perfil del alumno que los solicita dando respuesta a la consulta que plantea.
P2 ²	La herramienta de recomendación proporciona ejercicios relevantes y diversos entre sí para practicar la asignatura dando respuesta a la consulta que se plantea.
P2 ³	La herramienta de recomendación proporciona ejercicios relevantes para practicar la asignatura adaptados al perfil del alumno que los solicita.
P3	Con la herramienta de recomendación se puede aprovechar más el tiempo de estudio, al ayudar a localizar los ejercicios adecuados en cada momento.
Esfuerzo	
P4	La herramienta de recomendación me ha resultado, en general, fácil de utilizar.
P5*	He sido capaz de formular las consultas de manera relativamente fácil y rápida.
P6*	La estructuración de conceptos en la jerarquía es razonablemente lógica y en pocas interacciones he podido formular con rapidez mis consultas.
P7	La manera en la que se muestran los resultados de la recomendación es clara y comprensible.
P8	La cantidad de recursos recomendados en cada consulta eran suficientes.
Intención de uso	
P9	Sería interesante disponer de una herramienta como ésta en otras materias de conocimiento.
P10	Recomendaría la herramienta a otros alumnos que no la conocen.

¹ Para el prototipo de *KBPersonalization*.

² Para el prototipo de *KBdiversity*.

³ Para el prototipo de *KBnavigation*.

* Preguntas no aplicables al prototipo *KBnavigation*.

Figura 5.12: Interfaz de *KBPersonalization* y *KBdiversity*

Cada sub-objetivo ha sido refinado en varias preguntas (ver Tabla 5.3): 3 preguntas relacionadas con el rendimiento, 5 preguntas relacionadas con el esfuerzo y 2 preguntas relacionadas con la intención de uso. Estas 10 preguntas se reunieron en un cuestionario cuyas respuestas estaban definidas en una escala Likert de 5 puntos (1. Totalmente en desacuerdo – 5. Totalmente de acuerdo). Además, los participantes podían proporcionar comentarios de texto libre para explicar sus respuestas y para hacer comentarios sobre la herramienta. El análisis e interpretación de las respuestas y de los comentarios a estas preguntas nos han ayudado a evaluar los objetivos definidos.

Además de las preguntas relacionadas directamente con la evaluación se incluyeron otras dos preguntas para que se valorasen dos posibles mejoras en las estrategias de recomendación: la inclusión de explicaciones que indiquen por qué un objeto de aprendizaje ha sido recomendado y la posibilidad de realizar las consultas con ayuda de búsquedas de texto libre que faciliten la localización de los conceptos del dominio en la jerarquía de conceptos.

Para realizar esta evaluación y de acuerdo a los resultados obtenidos en las evaluaciones experimentales, se utilizaron tres prototipos. Cada uno de ellos ha sido desarrollado en Java y pensado para operar correctamente en sistemas operativos Windows, Mac y Linux. Cada uno de estos prototipos se corresponde con una de las tres estrategias diseñadas y presentadas anteriormente. Dichos prototipos siguen un modelo de operación cliente-servidor. El

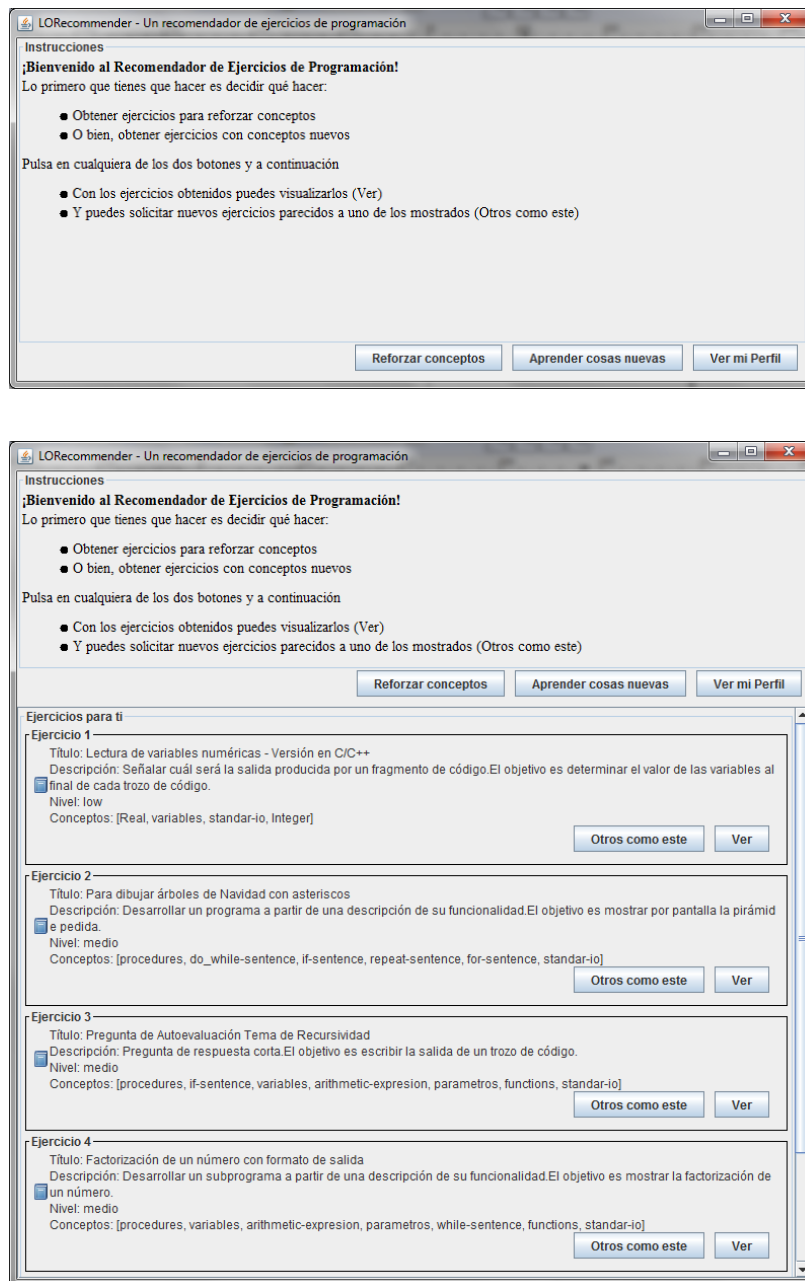


Figura 5.13: Interfaz de *KBnavigation* (arriba estado inicial, abajo resultado de una recomendación)

servidor contiene el repositorio de objetos de aprendizaje, la ontología del dominio y la información contextual. El cliente es una interfaz de usuario responsable de construir la consulta para el sistema, generar la recomendación y comunicarse con el servidor a través de la conexión a Internet para acceder a los objetos de aprendizaje recomendados contenidos en el repositorio. Las interfaces de la aplicación del lado cliente pueden verse en las Figuras 5.12 y 5.13. En la Figura 5.12 vemos la interfaz relativa a la estrategia de recomendación basada en conocimiento sensible al contexto que promueve altos niveles de personalización (*KBpersonalization*) y la estrategia de recomendación basada en conocimiento que promueve la diversidad (*KBdiversity*). En la interfaz de la Figura 5.12 el usuario primero debe elegir la estrategia a utilizar para, a continuación, seleccionar de la jerarquía de conceptos, que se encuentra en la parte superior, aquellas instancias o clases con las que quiere formar la consulta. Una vez que la consulta esté construida debe pulsar el botón “Recomendar” para obtener una recomendación. En la Figura 5.13 observamos la interfaz relativa a la estrategia de recomendación que combina proactividad y navegación por propuesta (*KBnavigation*). En esta interfaz el usuario debe seleccionar si desea “reforzar conceptos” o “aprender conceptos nuevos” y a partir de ahí obtendrá la primera recomendación. Para cada objeto de aprendizaje recomendado le aparece la opción “Ver” u “Otros como éste” que permite al usuario entrar en el proceso conversacional.

Para el primero de los prototipos, *KBpersonalization*, tal y como ya se ha indicado, se ha empleado la métrica de *calidad ponderada*, C_{1p} , por ser la que mejor resultados experimentales en cuanto a comportamiento obtiene. Para dicha métrica se ha fijado el parámetro α a un valor 0,2, por ser el valor que obtiene un alto compromiso entre la similitud con la consulta y la utilidad pedagógica asignada a cada objeto de aprendizaje de acuerdo a la información contextual del estudiante. El número de objetos de aprendizaje recomendados para una determinada consulta está fijado a $k = 7$ porque consideramos que es un número razonable de ejercicios entre los que el estudiante puede elegir sin llegar a sentirse sobrecargado de información.

En cuanto al segundo prototipo, *KBdiversity*, se ha empleado la métrica de *calidad armónica ponderada*, C_{2d} , por ser la que mayor ganancia de diversidad obtenía preservando altos niveles de similitud con la consulta. El parámetro α de dicha métrica está fijado a un valor 0,1, por ser el valor que obtiene un alto compromiso entre la similitud con la consulta y la diversidad del conjunto final. Finalmente el número de objetos de aprendizaje recomendados para una determinada consulta está fijado a $k = 7$.

El último de los prototipos, *KBnavigation*³, utiliza la métrica de *calidad*

³Para este prototipo hemos utilizado las siguientes clases incluidas en el framework: `LightAccurateRetrievalStage`, `ReachableConceptsFilter`, `WeightedMeanMetric` y `TopKSelectionStage`. Así mismo se ha implementado la clase `Navigation` que gestiona la navegación, tal y como se describió en la Sección 4.4.3.

ponderada C_{1p} con el parámetro $\alpha = 0,2$. La etapa de selección muestra al menos un representante de cada grupo en los que ha resultado particionado el repositorio siendo el número mínimo de objetos de aprendizaje recomendados $k = 7$. Si el repositorio resulta particionado en un número de grupos menor que k , entonces se completa la lista de recomendados con el siguiente mejor objeto de aprendizaje de cada grupo hasta completarla con k objetos de aprendizaje.

La evaluación tuvo lugar durante los cursos académicos 2011-12 y 2012-13. Tanto estudiantes como profesores participaron en sesiones formativas sobre el uso de los prototipos. A continuación tuvieron una sesión de 60 minutos para utilizar la herramienta de manera libre, en la que pudieron realizar distintas interacciones con el fin de obtener distintos objetos de aprendizaje relacionados con conceptos de Introducción a la Programación. Los profesores rellenaron los cuestionarios después de dicha sesión. Por su parte, los alumnos tuvieron a su disposición la herramienta durante 90 días más y después de estos días rellenaron los cuestionarios. La participación en la evaluación era voluntaria pero aquellos estudiantes que participaran podían obtener puntos extra de cara a la nota final de la asignatura. 14 profesores de asignaturas de Programación y 171 alumnos de los Grados de la Facultad de Informática de la Universidad Complutense de Madrid participaron en la evaluación. El grupo de profesores interactuó con los tres prototipos. Los estudiantes estaba repartidos en grupos y cada uno evaluó uno de los tres prototipos. El grupo que usó *KBpersonalization* estaba formado por 51 alumnos, 70 alumnos en el grupo del prototipo *KBdiversity* y 50 en el grupo que evaluó *KBnavigation*. A continuación pasamos a detallar los resultados para cada prototipo.

5.3.1. Resultados para la estrategia basada en conocimiento sensible al contexto que promueve altos niveles de personalización

En la Figura 5.14 podemos observar los resultados relativos al rendimiento de *KBpersonalization*. En general, existe un gran consenso sobre lo positivo que es el uso de la herramienta (P1). El 78 % de profesores y el 68 % de estudiantes creen que el recomendador ayuda a localizar recursos interesantes, que estaban adaptados al nivel de conocimientos del usuario y que además daban respuesta a la consulta que habían planteado (P2). Sin embargo, vemos diferencias entre profesores y estudiantes a la hora de valorar cuán rápido se ha conseguido esta localización (P3). Cerca del 28 % de los estudiantes puntuó esta afirmación con un 4 ó 5 reflejando así que habían podido aprovechar mejor el tiempo de estudio gracias a la herramienta, mientras que menos del 19 % de los estudiantes no estaba de acuerdo con esta afirmación. En cuanto a los profesores el 92 % puntuó con un 4 ó 5 dicha afirmación. De acuerdo a los comentarios proporcionados por los estudiantes,

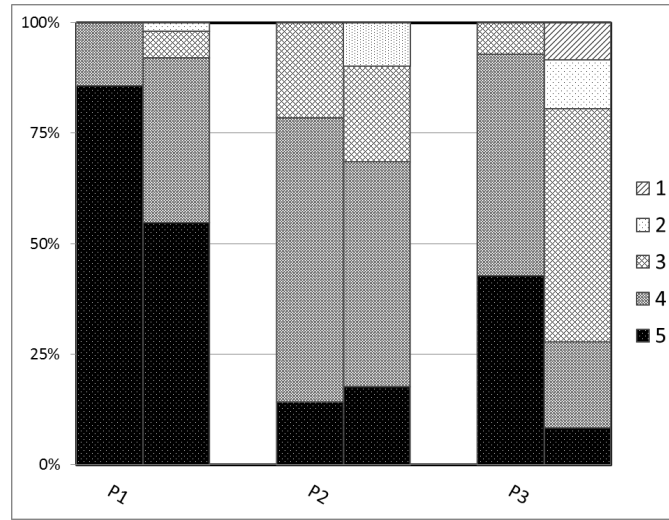


Figura 5.14: Frecuencias acumuladas para las valoraciones sobre el rendimiento (profesores a la izquierda, estudiantes a la derecha) para *KBpersonalization*

las respuestas negativas en P3 eran en gran medida debidas a la lentitud de la conexión a Internet y a la necesidad de familiarizarse con la jerarquía de conceptos.

Los resultados de las preguntas relacionadas con el sub-objetivo 2 (Figura 5.15) revelan que la herramienta de recomendación es, en general, fácil de utilizar. El 76 % de los estudiantes está de acuerdo con esta afirmación y todos los profesores que participaron en la evaluación la respaldaron (P4). En cuanto a la facilidad de realizar consultas al sistema, la mayor parte de los encuestados (92 % de profesores y 74 % de estudiantes) coincide en señalar que les ha resultado fácil proponer consultas al sistema (P5). Sin embargo, cuando se pregunta directamente por la jerarquía de conceptos (P6), vemos que el 62 % de los estudiantes señala que les ha resultado fácil de utilizar aunque cerca del 7 % coincide en señalar que han encontrado dificultades para localizar el concepto deseado en la jerarquía. Para el 71 % de los profesores y más del 76 % de los estudiantes la manera en la que se muestran los resultados de la recomendación es clara y comprensible (P7) y podemos ver que existen diferencias entre profesores y estudiantes a la hora de valorar la cantidad de recursos mostrados en cada recomendación (P8). Cerca del 59 % de los estudiantes cree que la cantidad de recursos mostrados en cada recomendación era adecuada (7 en este prototipo). En cambio menos del 14 % no está de acuerdo con esta afirmación. Sin embargo no explicaban si el número de objetos de aprendizaje era elevado o bajo. El 78 % de los profesores cree que el número de recursos mostrados tras cada recomendación es adecuado pero hay un 7 % que considera que 7 recursos es una cantidad baja, de

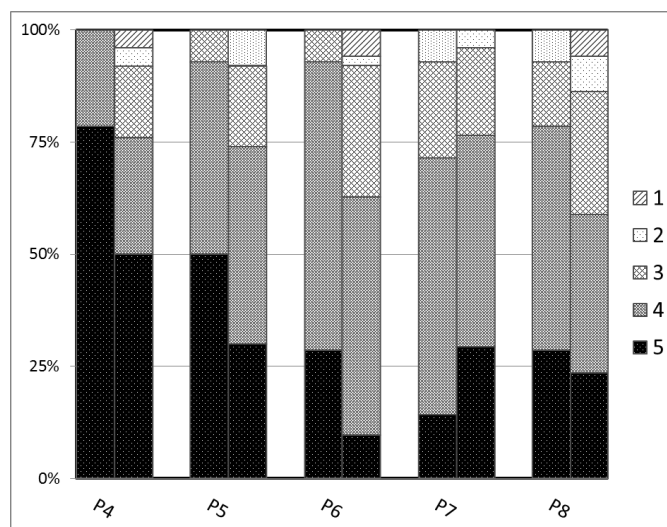


Figura 5.15: Frecuencias acumuladas para las valoraciones sobre el esfuerzo (profesores a la izquierda, estudiantes a la derecha) para *KBpersonalization*

acuerdo a los comentarios que proporcionaron después de la encuesta.

En cuanto a la intención de uso (sub-objetivo 3)(Figura 5.16), nos parece interesante señalar que más del 85 % de los profesores y más del 94 % de los estudiantes mostraron interés en disponer de una herramienta como ésta en otras materias de conocimiento (P9). Además una gran parte de ellos estarían dispuestos a recomendar el uso de estas herramienta a otros alumnos (P10).

En cuanto a las dos preguntas adicionales sobre la inclusión de posibles mejoras en la herramienta, el 80 % de los profesores y el 50 % de los estudiantes coinciden en señalar como positivo la inclusión de explicaciones. Por otra parte, más del 80 % de los profesores y del 70 % de los estudiantes estaban de acuerdo en incluir facilidades que guiasen en la localización de los términos en la jerarquía. Creemos que esta apreciación está relacionada con las dificultades para familiarizarse con la jerarquía de conceptos.

5.3.2. Resultados para la estrategia basada en conocimiento que promueve la diversidad

En la Figura 5.17 podemos ver los resultados para el sub-objetivo 1 del prototipo *KBdiversity*. En general, la herramienta también tiene un alto nivel de aceptación por parte de profesores y estudiantes (P1). Es en la pregunta P2 donde encontramos valoraciones más bajas para *KBdiversity* que para *KBpersonalization*. El 36 % de los profesores cree que la herramienta ha proporcionado ejercicios relevantes para el estudiante y que promovían la diversidad en el conjunto. Apenas el 15 % ha señalado estar en desacuerdo. Estos resultados mejoran si les preguntamos a los estudiantes, donde el 67 %

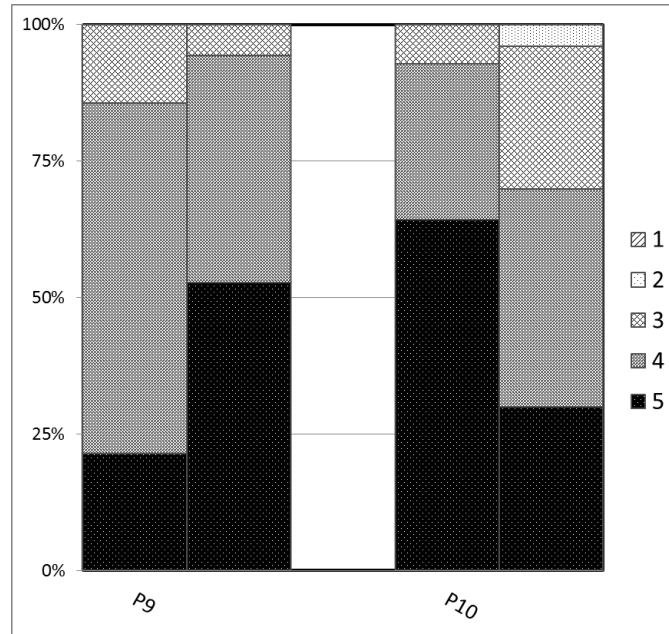


Figura 5.16: Frecuencias acumuladas para las valoraciones sobre la intención de uso (profesores a la izquierda, estudiantes a la derecha) para *KBpersonalization*

está de acuerdo con esta afirmación y sólo menos del 8 % en desacuerdo. Los comentarios proporcionados por los profesores nos hacen pensar que las bajas puntuaciones en P2 son debidas a la confusión que pueden generar algunas recomendaciones al obtener objetos de aprendizaje que reducen la similitud con la consulta en favor de ser más diversos. En cambio, los estudiantes no dijeron nada destacable sobre este hecho. En cuanto a la percepción en el aprendizaje, P3, cerca del 41 % de los estudiantes afirma con un 4 ó 5 haber aprovechado mejor su tiempo de estudio al utilizar la herramienta y menos del 14 % no estaba de acuerdo con esta afirmación. Llama la atención que el 45 % de estudiantes han contestado de una manera neutral a esta pregunta. Suponemos que quizá muchos de estos estudiantes no son conscientes de su autoaprendizaje. El 21 % de los profesores afirma haber encontrado ejercicios adecuados que permitirían aprovechar mejor el tiempo de estudio, pero cerca del 36 % discrepa.

Más allá de los resultados anteriores, las respuestas relacionadas con el sub-objetivo 2 (Figura 5.18) revelan que más del 73 % de los profesores y más del 80 % de los estudiantes están de acuerdo con que la herramienta de recomendación ha sido fácil de utilizar (P4) y que tanto profesores como estudiantes han sido capaces de formular las consultas de manera relativamente rápida (P5). Al igual que en el prototipo anterior, cuando se pregunta directamente por el uso de la jerarquía de conceptos (P6), las valoraciones

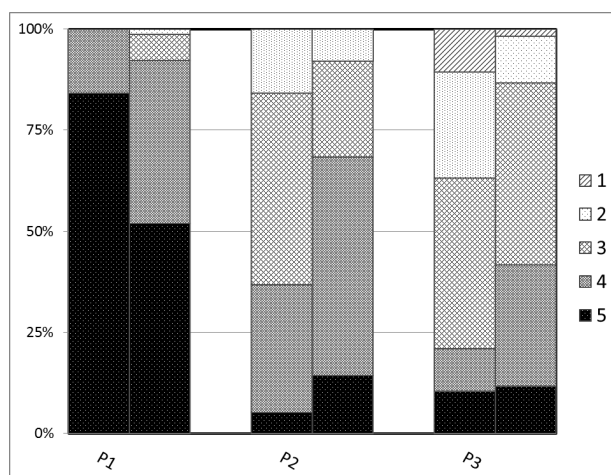


Figura 5.17: Frecuencias acumuladas para las valoraciones sobre el rendimiento (profesores a la izquierda, estudiantes a la derecha) para *KBdiversity*

descienden ligeramente, siendo el 68 % de profesores y el 62 % de estudiantes los que coinciden en señalar que ha sido fácil utilizar la jerarquía de conceptos. Los comentarios proporcionados por los estudiantes señalaban que algunos de ellos encontraban dificultades para relacionar los conceptos de la jerarquía con los temas del dominio de aprendizaje. Más del 78 % de los profesores y cerca del 65 % de los estudiantes están satisfechos con la visualización de los resultados (P7). Los comentarios proporcionados tanto por profesores como por estudiantes nos hicieron notar que las respuestas negativas en este apartado están relacionadas con la confusión generada por algunos títulos de ejercicios (muy similares) y con la propia naturaleza de algunos ejercicios. En algunos ejercicios se propone una solución a continuación del enunciado y los estudiantes señalaron que preferirían ver separada esta información. Existe menos consenso en cuanto a la cantidad de recursos mostrados (P8). El 52 % de los profesores señala que 7 es una cantidad adecuada de recursos, y menos del 16 % de ellos no está de acuerdo con la cantidad de recursos mostrados. En cambio más del 63 % de los estudiantes sí dice estar conforme con la cantidad de recursos mostrados.

La Figura 5.19 nos muestra los resultados para el sub-objetivo 3. El 79 % de profesores y el 90 % de estudiantes coinciden en afirmar que les gustaría disponer de herramientas similares en otras materias de conocimiento (P9). En cambio, para este prototipo, desciende el porcentaje tanto de profesores (menos del 58 %) como de alumnos (menos del 72 %) que recomendarían el uso de esta herramienta a otros alumnos que no la conozcan (P10). Pero, pese a ello, estos resultados muestran que la herramienta de recomendación tiene una gran aceptación tanto entre profesores como estudiantes.

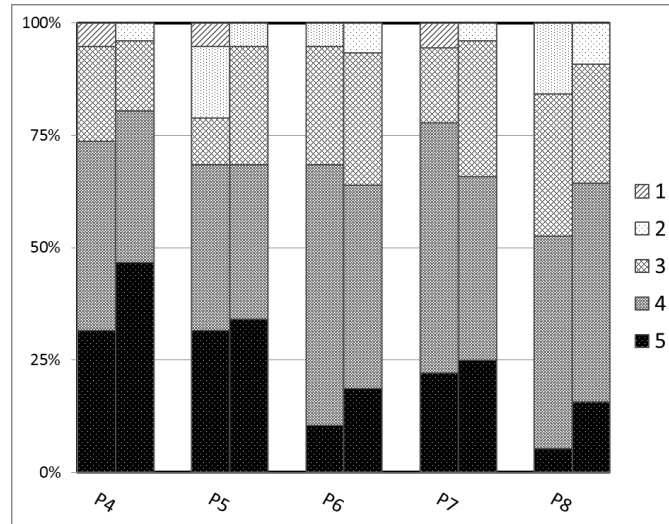


Figura 5.18: Frecuencias acumuladas para las valoraciones sobre el esfuerzo (profesores a la izquierda, estudiantes a la derecha) para *KBdiversity*

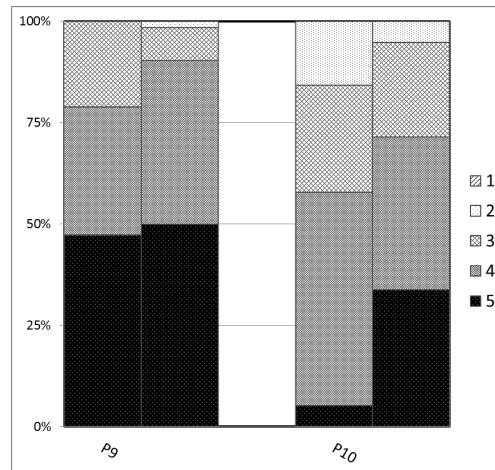


Figura 5.19: Frecuencias acumuladas para las valoraciones sobre la intención de uso (profesores a la izquierda, estudiantes a la derecha) para *KBdiversity*

Por último, las preguntas relacionadas con las mejoras en la estrategia se reparten los resultados de la siguiente manera. Cerca del 82 % de los profesores y del 55 % de los estudiantes consideran interesante incluir explicaciones sobre por qué un objeto de aprendizaje ha sido recomendado. El 83 % de profesores y estudiantes señalan como muy positivo la posibilidad de incluir un apoyo que guíe la localización de los conceptos a la hora de realizar consultas. Podemos observar que las repuestas son similares a las proporcionadas para el prototipo anterior aún siendo otro grupo de alumnos distinto. La mayor diferencia radica en el porcentaje de estudiantes que desearían contar con alguna facilidad de apoyo a la localización de términos en la jerarquía (en el caso de *KBdiversity* 83 % frente al 70 % en *KBpersonalization*). Estos resultados son naturales ya que este grupo de estudiantes es el que más dificultades ha encontrado para realizar la consulta y por lo tanto demanda de una manera más clara la posibilidad de incluir facilidades extra para la localización de los conceptos en la jerarquía.

Dejando a un lado las valoraciones referentes a los ejercicios propiamente dichos y que no interesan de cara a esta evaluación, las principales críticas a *KBdiversity* que realizaron los participantes nos llevaron a plantearnos cómo adaptar el nivel de diversidad de tal manera que los resultados siguieran siendo diversos pero sin resultar muy dispares entre sí, ya que algunos de los participantes, sobre todo profesores, se sintieron confundidos con algunas recomendaciones producidas durante la evaluación.

5.3.3. Resultados para la estrategia basada en conocimiento que combina proactividad y navegación por propuesta

Como observamos en la Figura 5.20 existe un amplio nivel de aceptación de *KBnavigation* tanto por profesores como por estudiantes (P1). También afirman (71 % de profesores y 60 % de estudiantes) que la herramienta de recomendación ha proporcionado ejercicios interesantes y adaptados al nivel de conocimiento (P2). Sólo menos del 9 % de los estudiantes y del 7 % de los profesores no estaban de acuerdo con esta afirmación. La pregunta P3 muestra que el 64 % de los profesores cree que la herramienta puede ayudar a los estudiantes a aprovechar mejor el tiempo de estudio. Y el 51 % de los estudiantes cree que han podido aprovechar mejor el tiempo de estudio gracias al uso del recomendador. Apenas un 12 % de estudiantes no está de acuerdo con esta afirmación.

Podemos observar en la Figura 5.21 que *KBnavigation* ha resultado realmente fácil de utilizar tanto a profesores como a estudiantes (P4). Para este prototipo las preguntas P5 y P6 relacionadas con el sub-objetivo 2 no eran aplicables, al carecer éste de consulta. La totalidad de profesores y el 85 % de los estudiantes están satisfechos con cómo se muestran los resultados de la recomendación (P7). El 64 % de los profesores y el 67 % de los estudiantes creen que la cantidad de recursos recomendados eran suficientes (P8). Sin

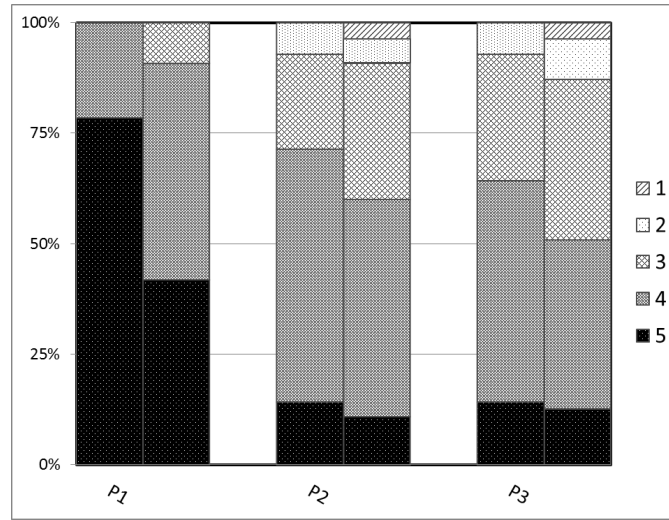


Figura 5.20: Frecuencias acumuladas para las valoraciones sobre el rendimiento (profesores a la izquierda, estudiantes a la derecha) para *KBnavigation*

embargo, los restantes participantes (profesores y estudiantes) no mostraban un total acuerdo con esta afirmación. En posteriores comentarios, los participantes indicaron que convendría incluir algún objeto de aprendizaje más en la recomendación.

En la Figura 5.22 tenemos los resultados para el sub-objetivo 3. El 85 % de los profesores y el 80 % de los estudiantes afirman que les gustaría disponer de herramientas similares en otras asignaturas (P9). Finalmente, más del 92 % de los profesores y más del 84 % de los estudiantes recomendarían a sus compañeros que utilizaran la herramienta (P10).

En cuanto a las preguntas sobre las posibles mejoras, más del 60 % de los estudiantes consideran interesante incluir explicaciones sobre por qué un objeto de aprendizaje ha sido recomendado.

Los comentarios proporcionados por profesores y estudiantes sobre *KBnavigation* sugerían, como mejora para el prototipo, introducir un filtro en las estrategias que pudiera regular tanto el tipo de ejercicio que se recomienda (ejemplo resuelto, ejercicio para practicar, pregunta de examen) como el nivel de dificultad del mismo (alto, medio, bajo). Estos comentarios serán tenidos en cuenta en el trabajo futuro.

5.4. Conclusiones

En este capítulo hemos presentado la aplicación de las estrategias de recomendación propuestas a un repositorio de objetos de aprendizaje de Pro-

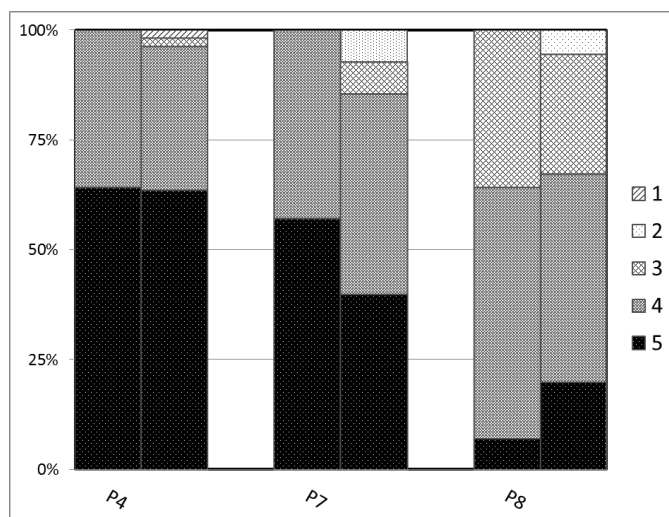


Figura 5.21: Frecuencias acumuladas para las valoraciones sobre el esfuerzo (profesores a la izquierda, estudiantes a la derecha) para *KBnavigation*

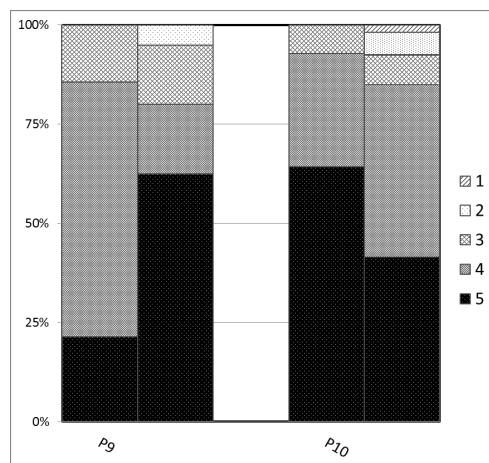


Figura 5.22: Frecuencias acumuladas para las valoraciones sobre la intencion de uso (profesores a la izquierda, estudiantes a la derecha) para *KBnavigation*

gramación. Posteriormente éstas han sufrido una evaluación compuesta de un análisis del comportamiento y una evaluación con usuarios reales.

El análisis del comportamiento sobre la primera de las estrategias (*KB-personalization*) sugiere que nuestra propuesta obtiene interesantes niveles de utilidad pedagógica mientras que preserva la similitud con la consulta. Este análisis también nos ha permitido valorar cuál sería el mejor valor para el parámetro α , empleado en las métricas de valoración de la calidad, que consigue equilibrar la calidad de la lista de recomendación con respecto a las dos relevancias. En cuanto a la estrategia que incluye diversidad en la recomendación (*KBdiversity*) hemos visto que obtiene interesantes niveles de diversidad en el conjunto de objetos de aprendizaje recomendados mientras que preserva la similitud de los mismos con la consulta, obteniendo además, gracias a la etapa de filtrado introducida en la estrategia, niveles interesantes de utilidad pedagógica. Al igual que en el caso anterior este análisis también ha servido para identificar el mejor valor del parámetro en la etapa de selección por diversidad que consigue un equilibrio en la calidad de la lista de recomendación.

Más adelante se ha presentado una evaluación con profesores y estudiantes para valorar la acogida de las estrategias de recomendación propuestas en el acceso a un repositorio de objetos de aprendizaje de Programación y el impacto percibido en algún aspecto del proceso de enseñanza-aprendizaje. En esta evaluación se han medido el rendimiento, el esfuerzo y la intención de uso futuro de las herramientas de recomendación desarrolladas siguiendo cada una de las estrategias. Esta evaluación ha revelado que tanto profesores como estudiantes creen que la herramienta de recomendación es un buen apoyo al proceso de enseñanza-aprendizaje. Además, consideran interesante disponer de este tipo de herramientas para repositorios de otras materias de conocimiento. Por otro lado, los resultados de las encuestas han ratificado los resultados obtenidos en el análisis del comportamiento. Profesores y estudiantes coinciden en señalar que los objetos de aprendizaje recomendados por *KBpersonalization* daban respuesta a su consulta (alto nivel de similitud) y se adaptaban a su nivel de conocimiento (utilidad pedagógica). En cuanto a *KBdiversity* profesores y estudiantes también coincidían en que los objetos de aprendizaje eran diversos entre sí dando a la vez respuesta a la consulta. Sin embargo, un 10 % de los profesores no estaba de acuerdo con esta afirmación, añadiendo que estos resultados podían despistar a algunos estudiantes. También ha sido buena la aceptación de la tercera estrategia *KBnavigation*. Profesores y estudiantes han coincidido en señalar que la herramienta proporciona ejercicios interesantes. En las respuestas de los cuestionarios también podemos observar que la curva de aprendizaje de uso de las herramientas es poco empinada. En *KBnavigation* la curva es todavía menor, debido a que no era necesario proporcionar una consulta. Sin embargo encontramos más desacuerdo en la cantidad de recursos que esta estrategia mostraba. Estos

resultados también nos han mostrado el interés de los usuarios (profesores y estudiantes) en una futura mejora en las estrategias reactivas: la posibilidad de realizar las consultas ayudados por facilidades de localización de los conceptos de la jerarquía. Algunos usuarios, sobre todo estudiantes, encontraron dificultades para encontrar en la jerarquía de conceptos aquellos que estaban buscando y aquellos que se correspondían con los estudiados en su curso de programación. Por otro lado, una gran parte de los usuarios coincidía en querer obtener una pequeña explicación sobre por qué un objeto de aprendizaje había sido recomendado.

En general los comentarios proporcionados tanto por los profesores como por los estudiantes en los distintos cuestionarios son muy positivos con respecto a las herramientas. Las principales críticas estaban relacionadas con algunas pequeñas erratas que contenían algunos ejercicios y con no disponer de un apartado en el que pudieran también realizar y enviar al profesor los ejercicios que realizaban. Por otro lado muchos de los estudiantes indicaron que en ocasiones la estrategia les volvía a recomendar ejercicios que ya habían realizado, siendo esto un comportamiento no deseable para ellos. Esto es debido a la ausencia de un filtro en la estrategia que decida cómo “descartar” aquellos objetos de aprendizaje que ya han sido recomendados anteriormente para dicho usuario. Esta cuestión la detallaremos en el próximo capítulo cuando hablemos de las líneas de trabajo futuro.

Capítulo 6

Conclusiones y trabajo futuro

*Sólo podemos ver poco del futuro, pero lo
suficiente para darnos cuenta de que hay
mucho que hacer.*

Alan Turing

En esta memoria de tesis se ha presentado una aproximación a la utilización de los sistemas de recomendación como soporte al acceso personalizado a recursos educativos existentes en repositorios electrónicos. Este proyecto se enmarca dentro del contexto investigador que aborda el traslado de las técnicas de recomendación al contexto educativo.

El primer resultado general de esta tesis ha sido la definición de tres estrategias de recomendación que hacen uso del conocimiento existente del dominio de aplicación, así como de la información contextual tanto del estudiante como de la actividad, con una representación semántica basada en ontologías. La primera de las estrategias presentada permite incorporar altos niveles de personalización en los procesos que producen la recomendación. La segunda estrategia presentada promueve la diversidad entre los objetos de aprendizaje recomendados a la vez que preserva los objetivos a corto plazo del estudiante, abordando así el problema de la sobreespecialización. La tercera y última estrategia combina proactividad y navegación por propuesta como modelo de interacción con el estudiante, alternativo a la naturaleza reactiva de las otras estrategias diseñadas, permitiendo recibir realimentación del usuario sin que suponga mucho esfuerzo para él. El objetivo de esta tercera estrategia es proporcionar una ayuda más adecuada a aquellos estudiantes que tienen una actitud pasiva o que tienen poco conocimiento del dominio como para proponer una consulta al recomendador.

Una vez definidas las tres estrategias propuestas se hacía necesario realizar una evaluación tanto computacional, que nos indicara las bondades de las mismas, como una evaluación con profesores y estudiantes que ratificaran la satisfacción con las mismas. Para poder realizar estas evaluaciones fue nece-

sario realizar implementaciones de las estrategias. El desarrollo de prototipos para cada estrategia y la subsiguiente detección de aspectos comunes y ejes de variabilidad ha dado lugar a otro importante resultado de esta tesis: un framework orientado a objetos que facilita el prototipado rápido de sistemas de recomendación basados en conocimiento en el ámbito educativo.

Finalmente, se ha presentado la evaluación de estas tres estrategias desde el punto de vista de su comportamiento y han sido aplicadas y evaluadas por profesores y estudiantes en un repositorio de objetos de aprendizaje relacionados con la Programación. Estas dos evaluaciones constituyen otro de los resultados globales de esta tesis.

Seguidamente pasaremos a exponer más en detalle las principales conclusiones y aportaciones de dicho trabajo así como posibles líneas de continuación del mismo.

6.1. Conclusiones

Como indicábamos al comienzo de la presente memoria, en los últimos años ha crecido la cantidad de recursos que podemos encontrar en repositorios de contenidos educativos. Tal cantidad de recursos a los que se puede acceder provoca, en caso de no disponer del soporte necesario, que el usuario que quiere utilizar el repositorio sufra una sobrecarga de información dificultando así el proceso de toma de decisiones sobre qué contenido es el más adecuado para él. Para aliviar este problema se hace necesario el desarrollo de herramientas que faciliten la localización de aquellos recursos que sean de interés para los usuarios de estos repositorios. Se ha detectado que existen algunas características deseables que deben ofrecer estas herramientas: adaptar la localización al conocimiento individual del usuario, a los objetivos y/o a las preferencias del mismo, incluir diversidad entre los resultados y que la manera de interactuar con el repositorio sea fácil para el usuario.

El análisis sobre el estado del arte en los sistemas de recomendación de recursos educativos nos ha dejado ver las carencias que existen todavía en este campo. Primeramente, se observó que son pocos los sistemas que hacen uso de las estrategias de recomendación basadas en conocimiento. En un dominio tan complejo como el nuestro es interesante tener una representación de todos los factores que pueden influir (y de los que se puede sacar partido) en la recomendación. Esto nos ha llevado a hacer uso de una representación rica del conocimiento basada en el uso de una ontología que aglutina toda la información que interviene en el proceso de recomendación, los conceptos del dominio, los objetos de aprendizaje y la información contextual.

Por otro lado, hemos visto que no son muchos los recomendadores sensibles al contexto dentro del ámbito educativo, y aquellos trabajos que sí apuestan por introducir el uso de información contextual para obtener las recomendaciones usan geolocalización o información sobre la tarea que está

realizando el estudiante en un determinado momento, pero no sacan partido al conocimiento del alumno o a las guías de aprendizaje. En el trabajo aquí presentado proponemos ampliar el uso que se está haciendo de la información contextual introduciendo información sobre el estado cognitivo del estudiante e información sobre la actividad en función de cómo puede influir ésta en el aprendizaje del alumno. En este punto el trabajo aquí presentado supone una notable aportación al campo.

También pudimos observar que no existían trabajos de recomendación en el dominio educativo que abordaran el problema de la sobreespecialización. Por lo tanto, muchos de estos sistemas adolecen de falta de diversidad en las recomendaciones lo que puede provocar la pérdida de interés por parte del estudiante con el uso del recomendador. Entre las estrategias propuestas dentro de este trabajo de tesis se ha adaptado y ensayado un algoritmo genérico de selección basado en diversidad, probado satisfactoriamente en otros dominios, con el fin de introducir diversidad en las recomendaciones propuestas.

Finalmente, la última carencia detectada estaba relacionada con la inclusión de mecanismos de interacción que facilitaran el acercamiento del usuario al uso del sistema. En particular, no había trabajos que ensayaran mecanismos de navegación por propuesta, que tan buenos resultados estaban mostrando en otros dominios.

Estas carencias de las herramientas de acceso a repositorios de recursos educativos son reconocidas por sus usuarios, quienes demandan soluciones a las mismas. Las tres estrategias basadas en conocimiento aquí presentadas abordan estas carencias y representan un avance en la solución de las mismas y, en general, en el estado del arte de la recomendación aplicada al ámbito educativo.

Cada una de estas tres estrategias da respuesta a una de las carencias detectadas anteriormente intentando remediar las necesidades que los usuarios de este tipo de repositorios echaban en falta y las limitaciones de los trabajos previos analizados. Los principales aspectos de estas estrategias son:

1. Se benefician de una descripción de los recursos educativos que aprovecha el potencial de los estándares de metadatos (concretamente IEEE LOM) y los enriquece a través de una indexación semántica basada en ontologías. En este sentido, coincidimos con otros autores sobre el uso de descripciones semánticas con el fin de facilitar el descubrimiento de contenidos educativos digitales. Las descripciones semánticamente enriquecidas de los recursos y una ontología que reúne tanto el vocabulario como las relaciones entre los términos del vocabulario son cruciales para localizar los recursos que satisfagan las necesidades de los usuarios.
2. Utilizan la información contextual sobre los usuarios. En los últimos

años, la información contextual ha conseguido un papel destacado en el dominio de recomendación. Algunos trabajos han comenzado a explorar los beneficios de utilizar este tipo de información para mejorar la calidad de las recomendaciones. Nuestro trabajo se suma a esta nueva tendencia en la investigación con el fin de encontrar contenidos educativos que coinciden con un contexto específico utilizando el conocimiento del estudiante. El uso de este contexto hace posible encontrar recursos educativos que son apropiados para el nivel de conocimiento del estudiante que realiza la búsqueda.

3. Utilizan la información contextual sobre la actividad. La ontología nos ha permitido incluir relaciones de precedencia entre los conceptos del dominio con las que establecer itinerarios o caminos de aprendizaje que son utilizados como información contextual. Esta información sirve para aumentar el nivel de personalización de las recomendaciones ya que se utiliza como un criterio en el proceso de recomendación que descarta aquellos recursos que no son adecuados para el estudiante actual.
4. Exploran un algoritmo de selección basado en diversidad incluyendo así variedad de recursos en las listas de recomendación propuestas. Esta diversidad permite que un estudiante no se vea desanimado al encontrar recursos muy similares entre sí, al mismo tiempo que maximiza los objetivos pedagógicos que un determinado estudiante puede alcanzar en una sesión de estudio.
5. Exploran dos modelos de interacción con el usuario. El primero de ellos es un modelo reactivo que impone que el estudiante tenga que realizar una consulta al sistema a partir de los conceptos del dominio que quiere aprender. Este modelo es favorable para aquellos estudiantes altamente motivados en el aprendizaje y que ya tienen un conocimiento del dominio que les permite identificar qué conceptos son los deseados. El segundo modelo de interacción propuesto explora una navegación por propuesta que propone al estudiante un conjunto de recursos sin que éste tenga que realizar ninguna consulta. Este modelo es más apropiado para aquellos alumnos con una actitud pasiva que pueden verse desanimados a usar el sistema por el mero hecho de tener que realizar una consulta. También es adecuado para aquellos estudiantes que todavía no tienen un conocimiento suficiente del dominio y les cuesta identificar qué conceptos son los que les gustaría aprender.

A modo de resumen, las aportaciones principales concretas de este trabajo de tesis son:

- Se ha presentado una revisión del estado actual en la investigación sobre los sistemas de recomendación. Esta revisión nos ha permitido

establecer cuáles son las principales características de estos sistemas, sus beneficios y algunos de los principales problemas que nos podemos encontrar en ellos. El resultado ha sido un marco teórico de caracterización de los sistemas de recomendación. Además se han analizado tres líneas de investigación abiertas en los sistemas de recomendación: la incorporación de información contextual en el proceso de recomendación, la inclusión de diversidad en las recomendaciones y la exploración de alternativas de interacción con el usuario que alivien la carga de trabajo para éste.

- Se ha realizado un análisis sobre el traslado de los sistemas de recomendación al ámbito educativo que ha permitido comprender cuál está siendo la aplicación de los sistemas de recomendación en este campo.
- En particular, se han analizado detenidamente los trabajos de recomendación aplicados a repositorios de recursos educativos y se han detectado las principales demandas de los usuarios de estos repositorios en cuanto a facilidades de acceso a los recursos se refiere. Fruto de este análisis se han considerado líneas interesantes de expansión las tres siguientes, cada una de ellas relativa a un aspecto de diseño recogido en el marco de caracterización de recomendadores mencionado anteriormente:
 - Inclusión de un mayor nivel de personalización haciendo uso de la información contextual y del conocimiento del dominio.
 - Incorporación de estrategias de selección que afronten el problema de la sobreespecialización y promuevan la diversidad en las propuestas.
 - Planteamiento de estrategias de interacción usuario-recomendador que supongan un menor esfuerzo por parte del usuario, como pueden ser estrategias proactivas y estrategias que permitan una navegación por propuesta.
- Se ha propuesto una estrategia de recomendación basada en conocimiento que hace uso de la información contextual para así aliviar los problemas de débil personalización encontrados en (Gómez-Albarrán y Jiménez-Díaz, 2009). Esta estrategia permite incorporar distintos niveles de personalización mejorando así la capacidad de adaptación al estudiante (Ruiz-Iniesta et al., 2009a,c, 2010). Esta adaptación se consigue explorando un modelo de personalización que incorpora la información contextual del estudiante y de la actividad al proceso de recomendación.
- Se ha propuesto una segunda estrategia de recomendación basada en conocimiento que promueve la diversidad entre los recursos educativos

recomendados a la vez que preserva los objetivos a corto plazo del estudiante (Ruiz-Iniesta et al., 2011a). En esta estrategia proponemos adaptar un algoritmo genérico de selección basado en diversidad, de tal manera que los recursos educativos recomendados sean los más similares a la consulta y al mismo tiempo los más disimilares con el resto de recursos educativos recomendados.

- Se ha propuesto una tercera estrategia de recomendación que emplea la navegación por propuesta como modelo de interacción con el estudiante (Ruiz-Iniesta et al., 2009b). Esta navegación engancha al estudiante en un proceso conversacional en el que podrá refinar el conjunto propuesto. El hecho de que la estrategia opere de manera proactiva puede ayudar a estudiantes poco motivados o que no tienen un conocimiento suficiente del dominio como para proponer una consulta, a que continúen su proceso de aprendizaje. Además gracias al uso de la información contextual –de estudiante y de actividad– y de las preferencias del estudiante recogidas en cada interacción se consigue un alto nivel de personalización en las recomendaciones.
- Se han identificado las necesidades de conocimiento específicas para estas tres estrategias, a saber: una ontología del dominio, los objetos de aprendizaje o recursos educativos, y la información contextual –de estudiante y de actividad. La ontología diseñada ha servido de punto de unión entre todos estos elementos ya que alberga una representación semántica de los conceptos del dominio, incluye una representación de los recursos educativos contenidos en el repositorio y ayuda en la representación de la información contextual. La representación de cada recurso educativo en la ontología relaciona el recurso con los conceptos del dominio que cubre. La información contextual de la actividad está representada mediante relaciones de precedencia entre conceptos que indican cuál es el itinerario de aprendizaje que un estudiante debe seguir para alcanzar los objetivos de estudio. Finalmente la información contextual del estudiante está relacionada con los objetivos alcanzados por el estudiante durante su proceso de aprendizaje. Estos objetivos están representados por conceptos que el estudiante debe conocer junto con el nivel de competencia o conocimiento alcanzado en cada uno de ellos.
- Se ha desarrollado un framework que permite el prototipado rápido de sistemas de recomendación basados en conocimiento para la recomendación de recursos educativos (Ruiz-Iniesta et al., 2011c, 2012a). Este framework supone una aportación al campo del desarrollo de sistemas de recomendación basados en conocimiento: no existían herramientas centradas en el diseño de este tipo de sistemas en el dominio de la enseñanza y además introduce un proceso de recomendación basado en

etapas que permite la configuración de múltiples sistemas de recomendación de manera sencilla.

- Se han aplicado las estrategias a un repositorio de recursos educativos ligados a la enseñanza de la Programación. Esta aplicación nos ha conducido a:
 - Desarrollar una ontología en OWL que contiene conceptos del dominio e itinerarios de aprendizaje sobre los mismos. Esta ontología puede ser reutilizada en el desarrollo de otros recomendadores en el mismo dominio.
 - Adaptar al estándar Learning Object Metadata (LOM) los recursos educativos de Programación contenidos en un repositorio existente.
 - Desarrollar un prototipo (con ayuda del framework desarrollado) para cada una de las estrategias propuestas que pudiera ser utilizado por estudiantes reales.
- Se ha realizado un análisis del comportamiento de las estrategias basadas en conocimiento propuestas que ha permitido estudiar la influencia del peso de las relevancias parciales utilizadas en el proceso de recomendación (Ruiz-Iniesta et al., 2011a). Este análisis ha mostrado que para las tres estrategias propuestas se obtienen interesantes niveles de utilidad pedagógica o de diversidad, según corresponda, mientras que preservan la similitud con la consulta. Este análisis se ha completado con una evaluación computacional sobre la calidad de las listas de recursos recomendados (Ruiz-Iniesta et al., 2011b) teniendo en cuenta el tamaño de las mismas y el orden en el que figuran los recursos recomendados. El resultado de esta evaluación computacional nos sugiere que en las tres estrategias, incluso con listas de recomendación pequeñas, se obtienen valores altos de personalización o diversidad, según corresponda, en las listas.
- Se ha realizado una evaluación de la aplicación de estas estrategias en una comunidad real de profesores y estudiantes en el dominio de la Programación para validar la aceptación de las mismas. En esta evaluación se han medido el rendimiento, el esfuerzo y la intención de uso futuro de las herramientas de recomendación desarrolladas siguiendo cada una de las estrategias. Esta evaluación ha revelado que tanto profesores como estudiantes creen que las herramientas de recomendación son de gran ayuda como apoyo a la utilización de un repositorio de recursos educativos. Tanto profesores como estudiantes señalaban que las estrategias proponían recursos adecuados al estudiante que los solicitaba. Y la mayoría de los participantes estaban de acuerdo en que les gustaría disponer este tipo de herramientas para repositorios de otras

áreas de conocimiento y que las recomendarían a otros usuarios. Los resultados obtenidos también nos han mostrado el interés de los usuarios (profesores y estudiantes) en una futura mejora de las estrategias: la posibilidad de realizar las consultas ayudados por facilidades de localización de los conceptos de la jerarquía. Algunos usuarios, sobre todo estudiantes, encontraron dificultades para encontrar en la jerarquía de conceptos aquellos que estaban buscando y aquellos que se correspondían con los estudiados en su curso de Programación. Por otro lado, una gran parte de los usuarios coincidía en querer obtener una pequeña explicación sobre por qué un recurso había sido recomendado.

6.2. Trabajo futuro

El trabajo descrito en esta memoria muestra el resultado final del mismo después de haber estudiado distintas estrategias de recomendación, de realizar su aplicación a un repositorio real de recursos de Programación y después de haber realizado una evaluación de las mismas. Pero existen aspectos que aún no han sido tratados y que se presentan como futuras líneas de continuación a partir del trabajo presentado.

A partir de la evaluación realizada con los usuarios (profesores y estudiantes) hemos detectado algunas líneas de mejora. La primera de ellas busca aumentar la confianza del usuario en el sistema. Para ello, puede resultar interesante que cada recurso educativo propuesto lleve asociada una explicación de por qué o en calidad de qué ha sido recomendado (conceptos de aprendizaje que cubre, dificultad del mismo, relación con otros recursos vistos, etc.). Por ello, como trabajo futuro dejamos la inclusión de explicaciones en las estrategias de recomendación propuestas. La primera explicación más evidente que surge es mostrar en los resultados de la recomendación, junto con el recurso recomendado, los conceptos que éste cubre para que el estudiante entienda mejor por qué el recomendador cree que esos son los recursos adecuados para él en el momento actual. Esta explicación puede ser incluida desde las primeras etapas del proceso de recomendación indicando por qué ha sido recuperado del repositorio dicho recurso, es decir qué conceptos cubre que hacen que sea recuperado, y puede verse mejorada a lo largo del proceso incluyendo información relativa a las propiedades del recurso educativo como por ejemplo, la utilidad pedagógica que éste muestra, la similitud con la consulta, etc. Si recordamos, los resultados de la evaluación con profesores presentada en el capítulo 5, nos sugerían que era necesario investigar cómo ayudar a los usuarios a percibir la diversidad en la lista de recomendaciones, y que así encuentren un mayor nivel de satisfacción y confianza con el sistema de recomendación. En la línea de lo comentado más arriba, estas explicaciones pueden servir para que los usuarios sean capaces de comparar entre los resultados propuestos mostrando las características

del producto que han hecho que haya sido elegido. De esta manera creemos que podemos ayudar a los usuarios a entender por qué las recomendaciones de la estrategia que promueve la diversidad en el conjunto recomendado son menos precisas con respecto a la consulta planteada y a que ellos entiendan la diversidad de las propuestas.

Otra mejora, en la línea de ayudar al usuario, está relacionada con la inclusión de mecanismos que faciliten la localización de los conceptos en la jerarquía. Esta mejora fue demandada, como ya vimos en los resultados de las evaluaciones realizadas, por una buena parte de los estudiantes que participaron en el experimento.

Para terminar el conjunto de mejoras relativas a la experiencia del usuario hemos dejado como trabajo futuro poder proporcionar comentarios por parte del usuario sobre si un objeto de aprendizaje recomendado le ha resultado útil o no. Cada usuario podría valorar un objeto de aprendizaje introduciendo una valoración junto con un comentario más desarrollado. Estas valoraciones permitirían introducir información social a las recomendaciones aportando un valor añadido en el proceso de recomendación. Esta mejora está relacionada con el modelo de gestión de objetos de aprendizaje mencionado en la Sección 3.2 y detallado en (Ruiz-Iniesta et al., 2012b).

Uno de los resultados del presente trabajo es la definición de un proceso de recomendación en cinco etapas que puede ser configurado para cada caso concreto mediante la instanciación de cada etapa. Además se ha desarrollado un framework para facilitar el desarrollo de sistemas de recomendación basados en este proceso de recomendación, pero todavía el framework presentado tiene un alto porcentaje de framework caja blanca. Sin embargo sería deseable que éste framework tendiera cada vez más hacia un modelo de caja negra. En este sentido el trabajo futuro más inmediato está relacionado con la inclusión en el framework de nuevas alternativas para distintas etapas del proceso de recomendación:

- Sería deseable explorar otras estrategias de recuperación de los objetos de aprendizaje del repositorio que hagan uso no sólo de los conceptos cubiertos por un determinado objeto sino también de otras características del mismo como puede ser el tipo de recurso o su dificultad. Esta información se encuentra reflejada en los metadatos del modelo LOM.
- Proponemos el desarrollo de nuevos filtros que hagan un mayor uso de la información contextual relativa a un estudiante. En particular, pretendemos incluir un filtro que descarte aquellos objetos de aprendizaje que un estudiante haya marcado como no relevantes (para ello podrían usarse los comentarios proporcionados por el usuario que hemos indicado más arriba).
- Las métricas de calidad que combinan métricas de utilidad individual y que se han propuesto en la valoración de los objetos de aprendi-

zaje pertenecen a la familia de las métricas de agregación básicas de la media cuasi-aritmética. Estas métricas de agregación controlan la importancia de las utilidades parciales modificando su peso (Adomavicius et al., 2011a). Sería interesante permitir la exploración del uso de otras funciones de agregación, tales como las empleadas para combinar preferencias individuales en recomendación para grupos (Masthoff, 2004).

- En cuanto a las métricas de utilidad individual queremos investigar la introducción de información contextual adicional que refleje el estilo de aprendizaje de un estudiante así como incluir información en los metadatos de un recurso acerca de los estilos de aprendizaje que promueve y explotar esta nueva información en las métricas de utilidad individual.
- En la etapa de selección, el trabajo futuro más inmediato se corresponde con la adaptación de otros algoritmos de selección basados en diversidad (como los presentados en el Capítulo 2), para poder evaluar las distintas estrategias resultantes y decidir qué algoritmo de selección obtiene mejores resultados.

Para completar el framework también está pensado incluir un conjunto de clases que ayuden a la evaluación automática de los sistemas de recomendación diseñados con él, facilitando así el proceso de evaluación de estos sistemas. En un primer momento se incluirán las mismas métricas de evaluación empleadas en este trabajo de tesis junto con ejemplos de prueba.

Por último, y relativo al modelo de interacción que explora un modelo de navegación por propuesta, la manera en la que se obtienen las preferencias del usuario es muy sencilla, pues el usuario simplemente indica su preferencia por un recurso frente a otro. Este enfoque lleva muy poca carga de realimentación, desde la perspectiva de los usuarios, pero, al mismo tiempo, proporciona una guía limitada en el proceso de recomendación. Así que pretendemos refinar la estrategia invitando a los usuarios a proporcionar información más precisa (por ejemplo, si la estrategia muestra las características por las que un recurso ha sido recomendado ésta puede pedir al usuario que seleccione la característica del recurso sobre las que de verdad tenga preferencia). Esta mejora podría reducir el ciclo conversacional y mejorar la precisión de las recomendaciones.

Finalmente, se propone estudiar el uso de este tipo de herramientas en otras disciplinas. Para el traslado de estas estrategias sería necesario analizar cuáles son las necesidades particulares de conocimiento y de información para estos dominios y cómo deben adaptarse las fuentes de conocimiento. Cada vez es más frecuente la existencia de repositorios de objetos de aprendizaje en muy variadas disciplinas que son susceptibles de aceptar herramientas de apoyo al usuario como las aquí presentadas.

Part II

Summary: Knowledge-based
recommendation strategies for
personalized access to learning
object repositories

Chapter 7

Introduction

In our everyday lives we are exposed to an amount of information that increases far more quickly than our ability to process it. So, it is time to develop technologies that alleviate this information overload. Research work on recommendation technologies helps to alleviate this burden by supporting users in pre-selecting information they may be interested in.

The development of electronic repositories for the storage of educational resources has been intensified during the last years and in most educational disciplines. The availability of these educational resources eases and motivates student self-learning as a complementary activity to lectures. However, the high number of resources that exist in these repositories makes the access difficult to those adapted to the individual knowledge, goals and/or preferences of the students. It is necessary to provide support for personalized searching functionalities, which retrieve resources that fit the needs, goals and preferences of the students. Hence, one of the goals of our research is to design recommendation strategies that support locating educational resources adapted to the student knowledge. Furthermore, this recommendation must be intended to propose a set of resources that are appropriate to the student so that she can take full advantage of a study session. It means that the proposals may not contain a lot of resources and it would be also desirable that the proposals be as varied as possible, in order to prevent the student get resources that are very similar among them. Finally this recommendation should explore mechanisms of interaction that allows to navigate through the space of resources and reduce the work load of the users. Overlook these needs can lead to underuse such repositories and, consequently, to loose the desired positive effect facing the learning process, despite the great effort involved in developing all the educational content gathered there (Dichev y Dicheva, 2012).

Research work on recommendation technologies helps to alleviate the aforementioned information overload by supporting users in pre-selecting information they may be interested in. Recommendation techniques come

in three basic flavours (Jannach et al., 2010): collaborative (or social) recommenders (Schafer et al., 2007), which generate recommendations for a target user based on the availability of ratings or preferences about the items; content-based recommenders (Pazzani y Billsus, 2007), which rely on item descriptions in order to make suggestions; and knowledge-based recommenders (Burke, 2000), which make use of additional, often manually provided, information about both the current user and the available items. Different hybrid recommendation strategies (Burke, 2002a) successfully combine individual approaches, such as collaborative and content-based techniques.

Although many works deal with the transfer of recommendation techniques into the educational domain, we found that these works do not make use of all of the particular characteristics that the learning domain imposes. In particular, we have seen that there are few studies that make use of available knowledge about the domain and about the student, reducing the recommendation techniques employed to recommend those resources that are popular among students. Using a rich representation of the knowledge can impact on achieving recommendations more adapted to the student and their needs. Hence, the three strategies presented in this thesis belong to knowledge-based recommender systems what represents an important contribution to the field.

On the other hand, in the educational domain, there is a wide range of information available about both users and the activity (learning goals, learning styles, mastery level of the student, etc..) that is not being exploited in these systems. Including all this information as a mean to refine the set of proposed resources will affect students' satisfaction with the recommender.

We cannot also forget that the list of recommended resources must be used to the student to make the most of a study session. This means that the recommendation lists may not contain a lot of resources and should not be overspecialised, they must not contain resources that are very similar among them. This could cause a lack of interest by students using the recommender. As far as we know, there are no previous works of recommendation in e-learning that face the overspecialization or lack of diversity problem. This is other novelty of this thesis work.

The last weaknesses identified in the work dealing with the transfer of recommendation techniques to the learning field is the lack of mechanisms for user-recommender interaction which ease the amount of student work. In this sense, we propose to exploit the benefits of a proactive interaction mechanism that guides the student through the resources to find those that best fit her needs.

The goal of our work on recommendation technologies in e-learning is to provide smart support for accessing to the Learning Objects (LOs) that exist in repositories. This proposal has lead to the definition of three recom-

mendation strategies that make use of existing knowledge of the domain, as well as additional information from both the student and the activity, with an ontology-based semantic representation. One of the strategies provides an student with a recommendation, list of educational resources that are adapted to the student's learning needs. The second one promotes diversity in the recommendation list. The third strategy explores a proactive model for user interaction based on a navigation-by-proposing model. The implementation of these three strategies has lead to the proposal of a developed framework for the rapid prototyping of knowledge-based recommenders to the learning field. These three strategies have been implemented and evaluated in a computational way and in a real learning field, where teachers and students have shown their satisfaction with the recommendation strategies designed. These three strategies presented will come to address the weaknesses identified in recommender systems in education.

Now that the major concepts of this thesis work have been stated, the following list enumerates its main goals:

- To conduct an in-depth analysis of the state of art in recommender systems. This analysis will allow us to establish key design aspects of recommender systems and what the main features of these systems are, their benefits and some of the main problems that we can find in them.
- To analyse some lines of current research on recommender systems: (a) the inclusion of diversity in the recommendation lists, (b) the incorporation of contextual information to improve the personalization of the recommendations and (c) the study of several models of user-recommender interaction.
- To conduct an analysis of how recommendation technologies has been transferred into the learning field and how the characteristics imposed by the learning field are taken into account in the design of recommender systems. For instance, successful learning paths and strategies could provide guiding principles for recommendation.
- To design recommendation strategies applied to the learning domain that make use of a rich knowledge representation and promote high levels of personalization, include diversity in the proposals and explore a model of navigation in the user interaction.
- To recognize and address the knowledge needs and its representation.
- To design a framework for the rapid prototyping of this kind of recommenders applied to the learning domain.

- To develop several prototypes that combine different approaches of the three proposed strategies in order to determine which combination achieves better results.
- To apply the proposed strategies into a particular learning domain such as the teaching of computer programming.
- To analyse the behaviour of the strategies in according to the quality of the recommendation lists proposed.
- To test the viability of the strategies designed in a real community of educators and students, who can give us information about the interest of the recommendation interface and the provided resources.

7.1. Outline

Next, we summarize the contents of the following chapters.

Chapter 2. Knowledge-based recommendation strategies for learning object repositories. This chapter describes the three recommendation strategies proposed that foster a personalized knowledge-based recommendation approach. These strategies employ contextual information about the students and their activities in order to generate recommendations, they include diversity in the recommendations or make use of a proactive navigation model to interact with the recommender.

Chapter 3. A framework for rapid prototyping of knowledge-based recommender systems in the learning domain. In this chapter we present a framework for rapid prototyping of knowledge-based recommender systems applied for learning object recommendation. The framework defines a recommendation scheme in five stages that can be configured and adapted to each concrete recommender. The framework not only provides implementations for alternative strategies for each stage, but also it can be enhanced with new implementations easily.

Chapter 4. Evaluation of the knowledge-based recommendation strategies. This chapter highlights the results of an experimental analysis of the strategies' performance. It also shows the warm welcome that educators and students gave to the application of the strategies to a repository of Computer Science Programming.

Chapter 5. Conclusions and future work. It summarizes the main contributions presented in this thesis. It also includes a discussion of future research lines and open challenges to improve the proposed approaches.

This document is a condensed translation of the original thesis in Spanish. Therefore, the correspondences between chapters are: This introduction summarizes the introduction and Chapter 2 of the Spanish version, Chapter 2 of this document summarizes Chapters 3 of the Spanish version. Chapters 3 and 4 summarizes Chapter 4 and 5, respectively. And, Chapter 5 corresponds to Chapter 6 in the Spanish version.

7.2. Publications

The following list presents the publications derived from this thesis:

1. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Recommendation in repositories of learning objects: A proactive approach that exploits diversity and navigation-by-proposing. En *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies*, páginas 543–545. IEEE Computer Society, 2009. ISBN 978-0-7695-3711-5.
2. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. User-adaptive recommendation techniques in repositories of learning objects: Combining long-term and short-term learning goals. En *Proceedings of the 4th European Conference on Technology Enhanced Learning: Learning in the Synergy of Multiple Disciplines*, páginas 645–650. Springer, 2009. ISBN 978-3-642-04635-3.
3. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Promoting strong personalization in content-based recommendation systems of learning objects. En *Proceedings of the XI International Symposium on Computers in Education*. 2009. ISBN 978-989-20-1774-7.
4. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Personalización en recomendadores basados en contenido y su aplicación a repositorios de objetos de aprendizaje. *IEEE-RITA*, vol. 5(1), páginas 31–38, 2010. ISSN 1932-8540.
5. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Combining personalization and diversity in a case-based recommendation strategy for the learning domain. En *Proceedings of the International Council for Educational Media and International Symposium on Computers in Education Joint Conference*, páginas 409–419. 2011. ISBN 978-972-789-347-8.
6. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. An experimental analysis of the behaviour of a personalized case-based recommendation strategy for the learning domain. En *Proceedings of*

- the 19th International Conference on Computers in Education*, páginas 135–137. National Electronics and Computer Technology Center, 2011. ISBN 978-616-12-0188-3.
7. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Un framework para el desarrollo de recomendadores basados en contenido aplicados a objetos de aprendizaje. En *Actas del 2º Taller sobre Ingeniería del Software en eLearning*, páginas 147–161. Universidad Complutense de Madrid, Área de Ciencias Exactas y de la Naturaleza, 2011. ISBN 978-84-694-7325-2.
 8. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. A framework for rapid prototyping of knowledge-based recommender systems in the learning domain. *Journal of Research and Practice in Information Technology*, vol. 44(2), páginas 167–181, 2012. ISSN 1443-458X.
 9. RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. A hybrid user-centered recommendation strategy applied to repositories of learning objects. *International Journal of Web Based Communities*, vol. 8(3), páginas 302–321, 2012. ISSN 1741-8216.

Chapter 8

Knowledge-based recommendation strategies for learning object repositories

Section 8.2 contains excerpts from Ruiz-Iniesta et al. (2009c); Ruiz-Iniesta et al. (2012b). Section 8.3 contains excerpts from Ruiz-Iniesta et al. (2009c); Ruiz-Iniesta et al. (2010). Section 8.4 contains excerpts from Ruiz-Iniesta et al. (2011a). Finally, Section 8.5 contains excerpts from Ruiz-Iniesta et al. (2009b).

In many different application domains, recommender systems have become essential tools to support users in pre-selecting information they may be interested in (Jannach et al., 2010). Particularly, in recent years the deployment of recommender systems has caused interest in the Technology-Enhanced Learning (TEL) domain, which aims to design, develop and test socio-technical innovations that will support and enhance the learning practices of both individuals and organizations (Manouselis et al., 2011b).

One of the first works that employed recommender systems in TEL domain was the one described in (Zaiane, 2002), where author used data mining techniques on the profiles of on-line learners in order to generate the recommendations. Other research works employed recommendation techniques to recommend courses and curriculum learning activities (Hummel et al., 2007; O'Mahony y Smyth, 2007), or to provide personalized and inclusive support in standard-based Learning Management Systems (Santos y Boticario, 2008a,c). Different recommendation approaches are also applied to adaptive navigation in educational systems. For example, the work described in (Tang y McCalla, 2003) details an evolving web-based learning system that can adapt itself not only to its users, but can also adapt its contents in response to the usage of the learning materials. Finally, several works

recommend exercises or learning materials –Learning Objects (LOs)– to the students. In (Bin Ghauth y Abdullah, 2009) the authors describe a combination of content-based and collaborative approaches based on the LO content and good learners’ rating. Michlík y Bieliková (2010) propose a method for personalized recommendation of assignments (tasks or exercises) in an educational system. To make the recommendations, each exercise in the course is evaluated and assigned a utility value according to three different criteria: concept appropriateness for the student, exercise difficulty and time period since the student’s last attempt to resolve the exercise.

The goal of our work on recommendation technologies in e-learning is to provide support for accessing to LOs that exist in repositories. We are conscious that the e-learning field imposes specific requirements on the recommendation process, for instance, taking into account the cognitive state of the learners, their goals and/or preferences or using pedagogic strategies as recommendation guiding principles. Additionally, avoiding the overspecialization problem that affects to the pure-similarity recommendation strategies (Smyth, 2007) and introducing diversity in the recommended LOs is also crucial for making the most of the recommendation session. Finally, this support should explore the use a mechanism of interaction that lets navigate through the space of resources. The research described in this thesis falls within the research area of knowledge-based recommendation techniques in the TEL domain that make use of contextual information. We present three recommendation strategies that deal with the LOs contained in digital repositories and makes use of contextual information about the activity and the user. Our activity-related contextual information refers to the inclusion of successful learning paths and strategies that could provide guiding principles for recommendation. Our user context is based on the learning goals, defined by a domain expert, that the learner will achieve during the learning process. As far as the overspecialization problem is concerned, we will profit from an existing diversity-conscious algorithm that, when adapted to our context, refines the order in which the retrieved LOs are finally proposed. In order to explore a user interaction model that eases the access to the repository we will present a recommendation strategy that, in a proactive manner, proposes LOs according to the student context. It exploits a kind of conversational technique that imposes little feedback overhead.

The chapter is organized as follows: Section 8.1 outlines the three context-aware recommendation strategies proposed. Section 8.2 details different knowledge sources required by the strategies, independently of the educational domain. In Section 8.3 we describe the first knowledge-based recommendation approach that provides a high level of personalization. Section 8.4 presents the recommendation strategy that promotes diversity among the recommender resources. Finally, Section 8.5 describes the recommendation strategy that exploits navigation-by-proposing.

8.1. Knowledge-based recommendation strategies for recommending learning objects

Our work aim to provide personalized recommendations that allow students to better access repositories of LOs. To that end the recommendation strategies proposed in this thesis belongs to knowledge-based recommenders (Jannach et al., 2010) and they rely on the existence of a knowledge base about the information sources involved in the recommendation process (LOs, domain ontology and contextual knowledge).

We propose three alternative knowledge-based strategies, each of them satisfies an essential requirement of recommenders of educational resources, (a) personalization, (b) overspecialization (or lack of diversity) and (c) facilities in the user interaction. Let's briefly introduce each strategy:

- The first knowledge-based recommendation strategy provides a high level of personalization (Section 8.3). This strategy follows a reactive approach: the student provides an explicit query and the recommender system reacts with a recommendation response. The student poses a query using the concepts existing in the domain ontology. This query represents her in-session or short-term learning goals: the concepts she wants to learn in the session. In this knowledge-based recommender, priority is given to those LOs that are most similar to the student's query and, at the same time, have a high pedagogical utility according to the student context information.
- The second strategy tries to avoid the overspecialisation problem that affects to the pure-similarity recommendation strategies (Section 8.4). This strategy includes diversity in the proposals making use of a ranking approach, called diversity-conscious ranking, inspired in diversity-conscious strategy described in (Smyth y McClave, 2001). Introducing diversity in the recommended LOs is also crucial for making the most of the recommendation session.
- The third strategy uses navigation-by-proposing, a simple conversational process that avoids posing direct queries and carries a small feedback overhead from the students' perspective (Section 8.5). Initially, the proactive recommendation result is a set of LOs presented as an assignment proposal to start with the learning activity. Priority is given to LOs that better adapt to the student context. The student can select one of the proposed LOs or she can ask for a refined proposal by entering in a conversational process. We use navigation-by-proposing due to its convenience in complex domains where the user may not be able to answer a given question because her domain knowledge is insufficient.

Section 8.2 provides a high-level description of the different knowledge sources used by the recommendation strategies, independently of the educational domain. Section 8.3 describes in detail the recommendation strategy that provides a high level of personalization. In Section 8.4 describes the recommendation strategy that promotes diversity in the recommendation. Finally, Section 8.5 presents the proactive strategy recommendation strategy that uses navigation-by-proposing.

8.2. The knowledge sources

In this section we first detail the knowledge sources employed by our recommendation strategies: the LOs (Section 8.2.1), the domain ontology (Section 8.2.2) and the contextual information (Section 8.2.3).

8.2.1. The learning objects

We describe the LOs using one of the most recognized metadata standards: IEEE LOM. LOM allows to tag the digital resources according to a set of predefined categories and assign values to each one. We propose to use a profile for describing LOs containing the next upper-level LOM categories: *Life cycle*, *Technical*, *Educational*, *Relation* and *General*.

The *Life cycle* category identifies the author and the licensing status of the LO. The *Educational* category helps to identify the type of LO (e.g., lecture note, solved example, quiz question, assignment, etc.). The *Technical* category groups the technical requirements and characteristics of the LO. Additionally, the *Relation* category identifies if an LO is a version of another one. Finally, the *General* category contains keywords that describe the domain learning topics covered by the LO according to the domain ontology vocabulary. So, these keywords are used to finally index the LO in the domain ontology, which is described next. While other category metadata can be exploited for retrieval purposes or only for presenting information to the user, the *General* category content will be especially important in our recommendation process because it will be used to compute the similarity between the LO and the learning goals that the user defines in a query.

8.2.2. The domain ontology

The ontology organizes the concepts that represent the domain topics using a taxonomy. It provides a general indexing scheme that includes similarity knowledge between the concepts representing these topics. As we will show, this similarity knowledge will be exploited by the recommendation strategy. The ontology also links each LO with the concepts that it covers. This information will be used by the recommendation strategy in order to

determine the suitability of each LO. As we will see in next subsection, it is also exploited to represent contextual information.

8.2.3. The contextual information

In this approach we propose the use of two contextual elements in the learning environment:

- **Activity context:** this knowledge is related to the inclusion of learning paths. A learning path reflects a successful sequence or order in which concepts are taught or learned in the corresponding field. Learning paths can help to filter out LOs that exemplify non-reachable concepts given the concrete cognitive state of the student. This contextual knowledge is static and specific to the learning domain. It must be defined by an expert (the instructor) before the students use the recommender. Learning paths are represented by a precedence property in the domain ontology.
- **Student context:** this knowledge concerns the goals achieved by the student in the learning process. The goals achieved are represented by the concepts that the student should know and the mastery level achieved in each of them. This level is considered to be a degree of satisfaction, a metric that allows the recommendation strategy to know about the student's knowledge level in a particular concept. This information evolves over time as long as the student interacts with the recommender and progresses in her learning.

We propose to use the activity context as a hard criterion to discard LOs that are not appropriate for the current student and the student context as a soft criterion that assigns utility to each LO.

8.3. A knowledge-based context-aware recommendation strategy that promotes high levels of personalization

The first knowledge-based recommendation strategy presented follows a process adapted to our special requirements in order to retrieve LOs in response to a query. This query is expressed in terms of the concepts that the student wants to practice during a learning session. Once the query is posed, the recommendation strategy obtains a recommendation list in a three-step process, retrieval, filtering and ranking, which are described next.

The retrieval process looks for an initial set of LOs that satisfy, in an approximate way, the student query. This process tries first to find the LOs indexed by the query concepts. If there are no LOs that satisfy this

condition, or if we are interested in a more flexible location, LOs indexed by a subset of the (same or similar) concepts specified by the student are retrieved.

Next, this initial set of LOs is filtered and only those LOs that cover ontology concepts “ready to be explored” by the student are finally considered in the ranking process. To do this, this process exploits contextual information about the domain concepts –activity context– and the cognitive state of the student –student context. The contextual information lets us know which LOs cover concepts that are not reachable by the student, in order to keep only the LOs that cover concepts “ready to be explored” by the student. We say that a concept is “ready to be explored” by a given student if, according to her current student context –the mastery level attained in the learning goals– and to the context about the domain –the learning paths defined in the ontology– it fulfills any of the following conditions:

- It is a concept already explored by the student, so that it appears in her student context with its corresponding competence level.
- It is a concept that the student has not explored yet but she can discover it: if a concept c_1 precedes a concept c_2 in the ontology, a student can discover c_2 if the student competence level for c_1 exceeds a given “progress threshold”. If several concepts c_1, c_2, \dots, c_k directly precede a concept c_x , the latter could be discovered if the student competence level in all the directly preceding concepts exceeds the given “progress threshold”.

In short, the goal of the filtering process is to discard LOs indexed by concepts non-reachable by the target student. This filtering process gives way to a light long-term personalization in this second phase of the recommendation strategy. This way, when two students pose the same query within a session but their subject masteries differ, the set of retrieved LOs could be different.

Finally, the ranking process uses a quality-based approach that fosters a strong personalized recommendation. Once the LOs are retrieved and filtered, the ranking phase sorts them according to the quality assigned to each LO. The quality is computed so that priority is given to those LOs that are most similar to the student’s query and, at the same time, have a high pedagogical utility according to the contextual information of the student.

In order to compute the quality of a given LO L for a student S that has posed a query Q , we will try two quality metrics defined as the weighted sum up of two relevancies: the similarity (Sim) between Q and the concepts that L covers, and the pedagogical utility (PU) of L with respect to the student

S :

$$Quality(L, S, Q) = \alpha \cdot Sim(L, Q) + (1 - \alpha) \cdot PU(L, S) \quad \text{where } \alpha \in [0, 1] \quad (8.1)$$

$$Quality(L, S, Q) = \frac{1}{\frac{\alpha}{Sim(L, Q)} + \frac{(1-\alpha)}{PU(L, S)}} \quad \text{where } \alpha \in [0, 1] \quad (8.2)$$

These quality metrics follow basic aggregation patterns frequently used in recommender systems (Beliakov et al., 2011).

In order to measure $Sim(L, Q)$, we take advantage of the similarity knowledge between domain concepts reflected in the domain ontology. The similarity $Sim(L, Q)$ between the concepts gathered in the query Q and the concepts that L covers requires to compute the similarity between two sets of concepts. A simplification consists of comparing the single concepts that result from the conjunction of each set of concepts –the concept that results from the conjunction of the query concepts (Q_conj_c) and the concept that results from the conjunction of the concepts covered by L (L_conj_c)– instead of directly comparing the two set of concepts. Assuming this simplification, we can use any accepted metric for comparing two hierarchical values. We decided to use a similarity metric successfully used (González-Calero et al., 1999):

$$Sim(L, Q) = \frac{|super(Q_conj_c) \cap super(L_conj_c)|}{\sqrt{|super(Q_conj_c)|} \cdot \sqrt{|super(L_conj_c)|}} \quad (8.3)$$

where $super(Q_conj_c)$ represents the set of all the concepts contained in the ontology that are superconcept of Q_conj_c and $super(L_conj_c)$ contains all the concepts within the ontology that are superconcept of L_conj_c . $Sim(L, Q)$ values lie in $[0, 1]$.

In order to measure the pedagogical utility $PU(L, S)$ we have adopted an instructional strategy that promotes filling the student's knowledge gaps by including remedial knowledge (Siemer y Angelides, 1998). The goal is to assign high pedagogical utility to L if it covers concepts in which the student has shown a low competence level. This way, L could help the student to improve her knowledge about these concepts and attain her long-term learning goals. We have decided to compute the pedagogical utility as follows:

$$PU(L, S) = 1 - NAM(L, S) \quad (8.4)$$

where $NAM(L, S)$ is the normalized arithmetic mean of the competence levels that the student S has shown in the concepts that L covers. $PU(L, S)$ values lie in $[0, 1]$. In short, (8.4) computes a low value of $PU(L, S)$ if the student masters the concepts that L covers. High values of $PU(L, S)$, on the contrary, are obtained if the student has a poor knowledge of the concepts covered by L .

Once the ranking stage calculates the usefulness of the LOs proposed by the previous stages, we sort them according to the quality assigned to each one and select those with higher scores. This way, the most interesting LOs for the target student (the *top-k* LOs) will ultimately be proposed.

The use of the student context information in the computation of $PU(L, S)$ and, as a consequence, in the computation of $Quality(L, S, Q)$ allows the inclusion of a considerable degree of personalization in the recommendation strategy. The final influence of the student context information in the recommendation results depends on the parameter α in Equations (8.1) and (8.2). Low values of α give priority to the pedagogical utility against the similarity to the query. In particular, $\alpha = 0$ represents the highest level of long-term personalization, and, in this case, the query (in-session goals) is used only in the retrieval stage. This ensures that the recommender system proposes LOs that meet the in-session goals at a minimum level, although the order in which they are proposed to the student is totally influenced by the long-term goals they let achieve.

8.4. A knowledge-based recommendation strategy that promotes diversity

The second strategy explores a knowledge-based recommendation strategy that combines the inclusion of personalization and diversity. On the one hand, and as far as the personalization aspect is concerned, this approach does not only promote the recommendation of LOs that satisfy the short-term learning goals –i.e., similarity with the student query– but it also promotes the recommendation of LOs that satisfy the long-term learning goals –i.e., fitting to the student cognitive state. On the other hand, and as far as the diversity aspect is concerned, we avoid LOs that are highly correlated among themselves. The aim is to recommend a restricted number of LOs where each LO covers slightly different goals in which the student can be interested in.

The knowledge-based recommendation strategy follows a reactive approach: the student provides an explicit query and the recommender system reacts with a recommendation response. Once the student poses the query, the recommended set of LOs is obtained in a four-step process: retrieval, filtering, rating and selection. Retrieval and filtering processes are similar to the ones employed in the previous strategy. The rating stage assigns

to each LO a relevance based on the similarity with the query (Equation 8.3). Finally, the selection stage will incorporate the diversity in the final set of recommended LOs. Next, we detail the selection process in this second strategy.

This selection step, called diversity-conscious selection, is inspired by a diversity-conscious strategy described in (Smyth y McClave, 2001): the Bounded-Greedy-Selection approach. The diversity-conscious selection algorithm sketches the ranking strategy adapted from the description of the Bounded-Greedy-Selection approach, where Q represents the query, R is the set of the LOs retrieved, filtered and ranked and k is the number of LOs finally recommended to the student:

Algorithm 2 Diversity-conscious selection algorithm

```

1: Input:  $Q, R, k$ 
2: Output:  $P$ 
3:  $P := \{\}$ 
4: for  $i := 1$  to  $k$ 
5: Sort  $R$  by  $Quality(L, P, Q)$  for each  $L$  in  $R$ 
6:  $P := P + \text{first}(R)$ 
7:  $R := R - \text{first}(R)$ 
8: end for
9: return  $P$ 

```

In short, diversity-conscious selection uses the set R obtained after the three previous processes in the strategy and it incrementally builds a set P of k LOs which will be finally proposed to the student. During each iteration, the remaining LOs in R are sorted by a diversity-aware quality metric and the LO with the highest quality is added to P and removed from R .

In order to compute the quality of a LO L , given the query Q and the set of LOs R different quality metrics can be chosen. We initially propose two quality metrics that combine two relevancies: the similarity (Sim) between Q and the concepts that L covers, and the diversity of L ($RelD$) relative to the LOs already selected to be recommended (P). These metrics are the following (they follow the same aggregation patterns used in (8.1) and (8.2)):

$$Quality(L, P, Q) = \alpha \cdot Sim(L, Q) + (1 - \alpha) \cdot RelD(L, P) \quad \text{where } \alpha \in [0, 1] \quad (8.5)$$

$$Quality(L, P, Q) = \frac{1}{\frac{\alpha}{Sim(L, Q)} + \frac{(1-\alpha)}{RelD(L, P)}} \quad \text{where } \alpha \in [0, 1] \quad (8.6)$$

The computation of the two partial relevancies *Sim* and *RelD* can follow different approaches and metrics. This way, the ranking model offers a framework which can be instantiated and result in different recommendation approaches. In order to compute the similarity we have adopted the same metric that in the previous strategy (Equation 8.3). As far as the relative diversity metric *RelD* is concerned, we adopted the one proposed in (Smyth y McClave, 2001):

$$RelD(L, P) = \begin{cases} 1 & \text{si } P=\{\} \\ \frac{\sum_{L_i \in P} (1 - Sim(L, L_i))}{|P|} & \text{otherwise} \end{cases} \quad (8.7)$$

Where $Sim(L, L_i)$ computes the similarity between the sets of concepts that the LO L and L_i cover.

8.5. A proactive knowledge-based recommendation strategy that exploits navigation-by-proposing

The goal of the proactive recommendation is to provide a more adequate help to students with a passive attitude or reduced domain knowledge (Ruiz-Iniesta et al., 2009b). The proactive recommender interface operates at the very start of the student's session. The proactive recommendation result is a set of LOs presented as an assignment proposal to start with the learning activity. Priority is given to the LOs that better adapt to the contextual information and, at the same time, meet the student preferences. The student can select one of the proposed LOs or she can ask for a refined proposal by entering in a conversational process.

On each recommendation stage, the repository is virtually divided into different clusters of LOs according to a well defined strategy and the recommender proposes LOs from the different clusters. On every stage the recommender chooses the LOs from the cluster which have the best pedagogical utility value for the student.

The navigation-by-proposing approach works in different stages, following the next strategy (students can stop the process at any step):

- During the first stage, the repository is divided, according to the contextual information, into two clusters of LOs: LOs that cover concepts already learned by the student (reinforcement) and LOs that cover new concepts that can be explored by the student (discover). Then the strategy retrieves two groups (one for each cluster) of LOs (the reinforcement and discover). To adapt the results to the current student and thus achieve a higher level of personalization, each of the sets of retrieved LOs is filtered and discarded those LOs that are not ready to

be explored. Here both sets are ordered according to a quality metric and is selected to least one member representing from each set. These LOs will be shown to the student.

- The second stage consists in a breadth-first traversal of the taxonomy of concept. Once the student has selected if she wants to reinforce or discover, the recommender creates a new proposal using LOs that cover concept from the top-most level. Every time the student decides to refine the proposal, the recommender will go down one level in the taxonomy according to the concept that the selected LO covers. The repository is divided in one cluster per ontology concept in the current level. If only one cluster is obtained, then the recommender will go down in the ontology until several clusters appear or until a leaf concept is reached. This strategy can be repeated until we reach the leaf topics in the ontology.

8.6. Conclusions

In this chapter we have described the development of three novel knowledge-based strategies for accessing educational repositories of LOs. Each one of these strategies tries to address one of these issues: the incorporation of contextual information in the recommendation process, the inclusion of diversity in the recommendation list and the use of proactive interaction mechanisms.

The first recommendation strategy feature is the inclusion of context information about learning domain and student knowledge. This context-aware approach provides new, personalized access to the LOs contained in educational repositories. The contextual information is responsible for the inclusion of learning paths to filter out the LOs that are less appropriate for the target student. The recommendation strategy also makes use of the student contextual information to assign utility values to each LO according to the cognitive state of the target student.

The second recommendation strategy focused in the inclusion of diversity in the recommendation list. The aim is to recommend a restricted number of LOs where each LO lets cover slightly different goals in which the student can be interested. As far as we know, there are no other works on recommendation in e-learning that face the overspecialization problem.

The last strategy combines a proactiv approach with navigation-by-proposing as a model of interaction with the student. This strategy addresses the last deficiency: the reactive nature of the strategy, which requires that the students had to perform a query. So, the strategy engages the student in a very simple conversational process that avoids posing direct questions, but presents recommendation alternatives and asks the student for preference-based feedback: the student shows a preference for an alternative over the

others.

Chapter 9

A framework for rapid prototyping of knowledge-based recommender systems in the learning domain

This chapter contains excerpts from (Ruiz-Iniesta et al., 2012a)

Our efforts in the deployment of recommendation strategies have focused on addressing three handicaps detected in recommender TEL systems: the weak personalization, the overspecialization and the inclusion of facilities in the user interaction model. The alternatives that we have proposed belong to the field of knowledge-based recommenders and they trust in the existence of a knowledge base with information about the sources involved in the recommendation process: the LOs, the contextual information and the ontology.

The analysis and comparison of our alternative recommendation strategies have imposed the development of multiple prototypes that implement them. In this context, we have searched for solutions for speeding up the development of prototypes of our recommendation strategies. Nowadays, there exist some libraries that help in the development of recommender systems, for example: SUGGEST¹, COFI², Apache Mahout³, Duine, MyMediaLite (Gantner et al., 2011), Crab⁴, GraphLab⁵ and Lenskit (Ekstrand et al., 2011). All of them are focused in the implementation of collaborative

¹ <http://glaros.dtc.umn.edu/gkhome/suggest/overview>

² <http://savannah.nongnu.org/projects/cofi/>

³ <http://mahout.apache.org>

⁴ <http://muricoca.github.com/crab/index.html>

⁵ <http://graphlab.org>

filtering algorithms, so they do not provide support for knowledge-based approaches. JColibri (Recio-García et al., 2008) is an alternative that supports the development of case-based recommender systems and it also has the basic tools needed for the development of knowledge-based and collaborative recommenders. However, JColibri focuses on the generation of recommendations based on similarity measures with the query, ignoring aspects of personalization, which are essential in LO recommendation. Consequently, none of these existing libraries adapted to the research area that we are exploring. Therefore, we decided to design our own framework that allows rapid prototyping of knowledge-based recommenders for the learning domain (Ruiz-Iniesta et al., 2011c, 2012a).

A framework is a reusable semi-complete architecture for developing applications in a specific domain, which allows code and design reuse (Pree, 1994). Rapid prototyping is an approach that allows to take crucial design decisions as early as possible. A rapid prototyping system should support maximal re-use and innovative combinations of existing methods, as well as simple and quick integration of new ones. Our framework has been designed within these principles. In this sense, our work represents a significant contribution in the field of frameworks for knowledge-based recommender systems.

This chapter describes the proposed framework and is organized as follows: Section 9.1 describes the recommendation stages of the recommendation process. Section 9.2 describes the framework for the LO recommendation that we have developed, detailing the main classes, the hooks and the default implementations provided for several strategies. Last section summarizes the chapter.

9.1. Stages of the recommendation process

The analysis of the structure of our recommendation processes for the learning domain concluded with the detection of a set of five aspects that adds variability and flexibility to the process. Each one of these aspects represent one stage of the recommendation process. In this way, our proposed recommendation process will consist of five stages related with the following questions:

- **How to make the user preference elicitation:** User preferences are commonly acquired using a query. This strategy could be called *reactive*, because the system reacts to a user query. Additionally, the recommendation process can start in a *proactive* way. In this case, the system takes the initiative and suggests a recommendation, for example, based on the information stored in the user profile. This aspect will be the first stage in the recommendation process that is called *query elicitation stage*.

- **How to retrieve the LOs stored in the repository according to the query:** This question is related with how the LOs are indexed in the recommender and which methods will be employed to retrieve the candidates resources for generating a recommendation. Although our current interests focus on exploiting the use of the ontology for indexing the LOs, we should not ignore the existence of another indexing alternatives. Regardless of how the LOs are indexed, we consider crucial a flexible retrieval process according to the terms included in the query. If there are not LOs that strictly satisfy the query proposed, this flexible retrieval suggests that the LOs indexed by a subset of the (same or similar) concepts in the query should be retrieved. The resulting stage from this question will be *retrieval stage*.
- **How to filter the retrieved LOs according to the user preferences:** The learning domain imposes new restrictions to the recommendation process related with the personalization. Although the users could have similar learning goals, the LOs that are useful for one of them can be useless for another one because they could have different knowledge about the domain or different learning styles. This question aims at the need of filtering the useless LOs according to the user profile, taking into account dimensions like her mastery level, previous interests or learning styles, among others. While this filtering process could be part of the retrieval stage, we prefer to consider it separately to add more flexibility to the resulting recommendation approaches. The resulting stage from this question will be *filtering stage*.
- **How to assess the quality of a LO:** A recommendation is generated using the most useful LOs for the target user, the LOs most similar to the query, etc. This usefulness is assessed defining a quality metric, which rate the LOs according to their attributes related, the query and the contextual information. A first approach proposes a similarity metric between the query and the LO, but we consider that the quality of a LO can be assessed according to different relevancies, like its pedagogical utility for the user, its correlation to the user interests, and so on. Moreover, the quality metric should not be limited to a unique relevancy, but the LO quality can be measured as an aggregation of different relevancies. This way, the analysis of the quality metrics does not only focuses on the selection of the individual relevancies employed to assess the utility of a LO. It is necessary to analyse the impact of the aggregation functions employed to combine these individual relevancies. The resulting stage from this question will be *rating stage*.
- **How to select the resources for the recommendation list:** Al-

though the retrieval and filtering process reduces the number of candidates to generate the final recommendation, this number is usually big enough to need a selection of the most interesting candidates to be recommended. We believe that the learning domain imposes a strong restriction on the size of the recommendation. Providing a user with a long list of LOs to practice can produce an overwhelming effect. Commonly, the way to shorten the recommendation is limiting the list to the k most useful candidates according to the quality metric. However, we cannot obviate that a short recommendation can be overspecialised, so the LOs included in the recommendations are highly correlated between them. This way, if the first LOs are not interesting for the user, probably none of the selected LOs will have any interest. For this reason, we consider that the inclusion of diversity approaches and its impact in the final recommendation is mandatory when the final candidates are selected. The resulting stage from this question will be *selection stage*.

9.2. A framework for learning object recommendation

The analysis about the variability aspects in a knowledge-based recommendation process has evolved in the definition of a framework for rapid prototyping of these recommender systems. First, as mentioned earlier, each of the issues discussed as variability shafts in Section 9.1 has been identified as a stage of the recommendation process. Subsequently, we have defined the abstract classes of the framework in charge of the control flow and responsible for executing the stages orderly. Next, we have defined the hooks of the framework (Pree, 1994), those predefined points that need to be configured and where subclasses and specific methods of the recommender must be implemented. Finally, we have created a set of classes that inherit from the abstract classes in this framework and implement some of the specific strategies used in our prototypes. Thus, the development of new recommender systems becomes much easier. The end result is the class diagram depicted in Figure 4.1 of the Spanish version.

9.2.1. The main classes of the framework

The main class of the framework is **KBRecommender**. This class contains the infrastructure for implementing a knowledge-based recommender of LOs. This class is also responsible for controlling the execution of the recommendation process. The configuration parameters of the recommendation process are stored following a blackboard architecture (Stegemann et al., 2007). Any recommender implemented using this framework needs to create at least a

subclass of `KBRecommender`. It has to implement the hooks that will be presented in section 9.2.2 or it can use some of the default implementations provided by the framework.

Each stage of the recommendation process corresponds to a framework class:

- The `QueryElicitationStage` class is responsible for creating the query that will be employed to start the recommendation process.
- The `RetrievalStage` class corresponds with the stage responsible for generating the set of LO candidates from the query.
- The `FilteringStage` class is responsible for making a first filtering on the set of candidates from the previous stage. Its behaviour is common to all recommenders: iterating over the set of LOs retrieved and deciding which LOs should pass to the next stage. Therefore, this stage delegates in a `Filter` object, according to the *Strategy* design pattern (Gamma et al., 1995). This object will decide whether a LO will be removed from the set of candidates.
- The `RatingStage` class is responsible for assigning quality values to each LO candidate after the filtering stage. Like in the `FilteringStage` class, the behaviour of the `RatingStage` class is the same in all recommenders: iterating over the set of LOs and assigning a quality value for each one. As in the previous case, we follow the *Strategy* pattern: this class delegates the quality assessment in an object that extends the `QualityMetric` (that will be detailed below).
- The `SelectionStage` class is responsible for building the final set of LOs recommended to the user.

The set of classes corresponding to the stages of the recommendation process is completed with the classes in which they delegate and classes that are used for transferring data between stages:

- The `Filter` is responsible to decide which LOs must be removed from the set. This class needs to have a defined filtering criteria that will be used in the method `boolean filter (LO)`. This method is invoked by the class `FilteringStage` and the return value indicates whether the LO must remain in the set or may not.
- The `QualityMetric` class has been designed taking into account that the assessment of usefulness should be a flexible algorithm. The LO quality can be measured with different utility metrics and it is also necessary to allow to combine, in a flexible way, several metrics to compose the LO quality. For this reason, we have adopted the *Composite* design

pattern (Gamma et al., 1995) to implement this class. This pattern establishes that a **QualityMetric** can be an individual utility metric (represented by the **Utility** class), or a composition of different quality metrics (represented by the **AggregationFunction** class). The latter has been improved including weights for each component that takes part in the aggregation function. The individual utilities are included in the aggregation by using the `add(QualityMetric, double)` method, which supports the use of weights for each individual metric in the global quality metric.

- The data transfer objects (Alur et al., 2003) **QueryTO**, **QualityTO** and **RecommendationTO** are generated by the classes **QueryElicitationStage**, **QualityMetric** and **SelectionStage** respectively. There is also a last data transfer object, **LO**, which is responsible for representing the information related to an LO. **LO** objects represent each LO in the repository. Each **LO** stores the necessary information for each LO to the recommendation process, such as ID, name, covering concepts, etc.
 - The **QueryTO** objects represent the query.
 - Although the quality value of an LO often takes a normalized value in the interval $[0, 1]$ we decided to use a transfer object called **QualityTO** as result of the assessment stage. This transfer object is generated by the **QualityMetric** class and it can contain, in addition to the quality value, all those attributes and values that may be relevant to the next stage and, therefore, to the final recommendation.
 - The same solution has been used in the **SelectionStage** class, which generates **RecommendationTO** transfer objects to store the LOs recommended and any additional data associated with each LO that will provide information about the recommendations. In this way, we combine in the assessment process two different approaches: the common numerical approach and the symbolic criteria. The latter can be useful, for example, when doing explanations about the behaviour of the recommender, what could increase the user confidence and trust in the recommendation process.

The last one is the **Blackboard** class. This class contains the initial state of the problem to solve, represented by a query and configuration parameters of the process (number of LOs recommended, user contextual information). The end result of the recommendation process must be written on the board as a solution to the problem.

9.2.2. The framework hooks

The classes described above have several hooks that modify the behaviour of the recommender. These hooks are instantiated by the subclasses that implement concrete strategies in order to extend the framework or to use it for building a customized recommender.

The classes that implement the recommendation process stages have been developed using a two-phase initialization pattern. This pattern guarantees that the objects can be efficiently used in consecutive executions of the recommender. This pattern imposes that any subclass that extends a class modeling a stage of the process must implement the following hook methods: `bool init()` and `void end()`. The former will be used to validate and initialize the stage, either by using the configuration parameters obtained from the blackboard, or by using the configuration parameters passed during the object creation. The latter will be mainly used to release resources and let the object ready for a new recommendation process. This two-phase initialization pattern is also used in other framework classes.

Some classes that represent the recommendation stages have another hook methods:

- The `QueryElicitationStage` class defines the `QueryT0 generate-Query()` hook method. It is responsible for building the query employed in the recommendation process.
- The `RetrievalStage` class defines the `Collection<LO> retrieve(QueryT0)` hook method. This method returns a set of LOs using the query generated by the previous class.
- The `SelectionStage` class defines the `List<RecommendationT0> select(Collection<QualityT0>)` hook method. This method generates the resulting list of recommended LOs. This method builds the `RecommendationT0` objects for each LO using a list of LOs and their respective `QualityT0s` objects, generated by the `RatingStage` class.

The `FilteringStage` class has a `Collection<LO> filter(Collection<LO>)` method to filter the retrieved LOs. However, this method does not represent a hook method. As mentioned before, the flexibility of this stage is provided by the delegation of the filtering strategy in the `Filter` class. This class does have a hook, the `bool filter(LO)` method, responsible for deciding when an LO is removed from the list of candidates passed to the rating stage. The `Filter` class has also the methods `bool init()` and `void end()` which are imposed by the two-phase initialization pattern.

Like the previous class, the `RatingStage` class has a `Collection<QualityT0> rate(Collection<LO>)` method, which is responsible for assessing the LOs, but it does not represent a hook. This method is responsible for it-

erating over the retrieved and filtered LOs and it delegates the computation of the quality value of an LO to the `QualityMetric` class. Then, it composes the collection of `QualityTOs` that will be transferred to the next stage of the recommendation process. The `QualityMetric` class has a hook method `QualityTO computeQuality(LO)`. Both the individual utility metrics (`Utility`) and aggregation functions (`AggregationFunction`) define their behaviour by implementing this method, which is responsible for calculating the quality of the LO provided as parameter and generating the associated `QualityTO` object. Like in the `Filter` class, the use of the two-phase initialization pattern imposes two additional hook methods in the `QualityMetric` class: `bool init()` and `void end()`.

The `KBRecommender` class has a set of hooks that have to be implemented by any subclass in order to generate a recommender, namely:

- `QueryElicitationStage createQueryElicitationStage(), RetrievalStage createRetrievalStage(), SelectionStage createSelectionStage()`: These are the *Factory Methods* (Gamma et al., 1995) responsible for creating the concrete instances of the query elicitation, the retrieval and the selection stages, respectively. It is worth noting that there is no factory method for the rating (`RatingStage`) and filtering stages (`FilteringStage`) because, as we detailed above, other classes are responsible for providing the flexibility of these stages.
- `Filter createFilter()`: This factory method is responsible for creating the object of the `Filter` class used by the `FilteringStage` class to discard the useless LOs.
- `QualityMetric createQualityMetric()`: This factory method is responsible for creating the `QualityMetric` used by the `RatingStage` class to assess the LOs. This method creates the objects that conform the `QualityMetric`. When using a quality metric as a combination of multiple utilities, these are composed using a subclass of `AggregationFunction`. The individual utilities are included in the aggregation using the `add (QualityMetric, double)` method, which supports the use of weights for each individual metric in the global quality metric.
- `void configureRecommendation()`: It sets the parameters required for a recommendation, keeping them on the blackboard. A user identifier is the unique mandatory parameter needed for a recommendation, which can be used to access to the user profile in those stages that need it. However, this method could be implemented so that it add on the blackboard all those additional parameters that are considered necessary for the specific recommender.

- `void finishRecommendation(List<RecommendationT0>)`: This method is in charge of processing the result of the recommendation. This may include, among other issues, displaying the recommendation, the data storage for further evaluations of the recommender or the update of the user context according to the recommendation provided or the way the user interacts with this recommendation.

In order to coordinate the execution of the hooks described along this section, the `KBRecommender` class implements the following *Template Methods* (Gamma et al., 1995):

- `bool init()` This method is responsible for initializing the recommender. It creates the instances that implement the recommender stages, the filtering strategies and the quality metrics by using the factory methods described above, as shown in Figure 4.2 of the Spanish version.
- `bool initRecommendation()`: This method is responsible for preparing the execution of a recommendation. First, it invokes the method `configureRecommendation()`. Then, it initializes the recommendation stages by invoking the respective `bool init()` method of each class, as shown in Figure 4.3 of the Spanish version.
- `List<RecommendationT0> recommend()`: This method is responsible for generating the recommendation. It executes each of the stages of the recommender in order, invoking the hook methods of every class that serves as a recommender stage, as shown in Figure 4.4 of the Spanish version. The final result is an ordered list of `RecommendationT0` objects (each one encapsulates a LO and the corresponding additional information).
- `void endRecommendation(List<RecommendationT0>)`: This method is responsible for the final processing of the recommendation and the release of the recommender stages. The former is implemented by calling the `void finishRecommendation(List<RecommendationT0>)` method, while the latter is implemented by sending the `void end()` message to each object representing a recommender's stage, as shown in Figure 4.5 of the Spanish version.

9.2.3. The implementation of some concrete classes

In order to tend to a black box model, we have included the implementation of some subclasses of the abstract classes. This way, framework users can build, in an easily way, basic recommender systems. Next, we summarize some of these concrete classes and in (Ruiz-Iniesta et al., 2011c, 2012a) we can find the complete implementations.

For the query elicitation stage the framework includes the implementation of the **UserQueryElicitationStage** class, which inherits from the **QueryElicitationStage** class. The method **QueryTO generateQuery()** of this subclass uses a graphical interface to interact with the user and to request the query. This interface shows the available domain concepts in the query ontology and the user selects the concepts that conform the query. Then, we have added two more classes that generate queries based on the contextual information regarding of the current user and the contextual information on the activity. These classes are **ReinforcementConceptsQuery** and **DiscoverConceptsQuery**. **ConceptsQueryElicitationStage** is responsible for generating a query that contains a set of concepts obtained through the blackboard. **FromFileQueryElicitationStage**, **CmdLineQueryElicitationStage** y **RandomQueryElicitationStage** are included in order to allow the evaluation of the recommendation strategies.

For the retrieval stage we have included the implementation of different strategies. On the one hand, we have implemented an accurate retrieval strategy –**AccurateRetrievalStage**– that is responsible for retrieving those LOs indexed, at least, all the query concepts. On the other hand, we have implemented an approximate retrieval strategy –**ApproximateRetrievalStage**– that selects those LOs indexed at least one query concept or by siblings of one or several query concepts according to the concept. Finally **LightAccurateRetrievalStage** selects those LOs indexed with at least one query concept. All classes inherits from the **RetrievalStage** class.

For the selection stage we have developed a top k selection strategy –the **TopKSelectionStage** class. This class sorts the set of candidates in terms of the quality assigned to each LO, and then select the k best LOs. The parameter *k* is provided by the developer and it may be defined when configuring the recommender. We have implemented the **RelDiversitySelection** class which inherits from the **SelectionStage** class. The select method implements the algorithm proposed in Section 8.4.

The framework includes a filter that relies on the existence of learning paths in the ontology. These learning paths and the competence level attained by a given student for each concept allows the identification of the attained concepts –concepts already explored by the student–, the concepts that the student can learn –concepts ready to be discovered– and the concepts that the student is not ready to learn yet –unreachable concepts. Using this classification, the filter discards from the retrieved set those LOs that cover unreachable concepts for the target student. The **ReachableConceptsFilter** class, that inherits from the **Filter** class, implements this strategy through the **bool filter(LO)** method, which checks whether an LO covers any unreachable concept.

Finally, we have implemented several individual quality metrics and aggregation functions that combine the individual quality metrics in order to

provide complex quality metrics.

The first quality metric implemented assesses the utility of each LO based on the similarity between the concepts that the LO covers and the query concepts. We decided to employ the quality metric that we previously defined in (González-Calero et al., 1999). This metric has been implemented in the `QuerySimilarity` class that inherits from the `Utility` class.

Another individual quality metric measures the pedagogical utility (PU) that an LO L has for a student S . `PedagogicalUtility` class, which inherits from the `Utility` class implements this metric.

Regarding to the aggregation functions, we have included the `Weighted-MeanMetric` and `HarmonicWeightedMeanMetric` classes, which extend the `AggregationFunction` class within the framework. The former computes the utility as the weighted mean of all the utilities that conform the quality function. The latter combines the utilities using the harmonic weighted mean metric. Both aggregation functions force the recommender developer to provide the weight of each individual utility in the resulting quality metric.

9.3. Conclusions

In this chapter, we have presented a framework that allows us to build prototypes of knowledge-based recommender systems of LOs in an easy way. We have identified five stages in the recommendation process, each one considered to be an aspect of variability. The framework was designed so that the developer can easily create recommenders that use alternative strategies, implemented for each stage. It was also considered that the framework can easily be increased by implementing new strategies.

Chapter 10

Evaluation of the knowledge-based recommendation strategies

Section 10.2.1 contains excerpts from Ruiz-Iniesta et al. (2011b,a).

Traditionally, the study of a recommender system focuses on measuring the accuracy of the system in predicting users' preferences (Herlocker et al., 2004). As stated (Herlocker et al., 2004), the research community has moved from the *annotation in context* task (i.e., predicting ratings) to the *find good items* task (i.e., providing users with a ranked list of recommended items), which better corresponds to realistic settings in working applications where recommender systems are deployed. As a result, a large amount of recent work has focused on evaluating top-N ranked recommendation lists (Shani y Gunawardana, 2011).

In case of recommenders of learning resources, it is more important assessing the impact of these resources on the student learning and the student satisfaction than measuring the recommendation accuracy. It is necessary to establish alternative ways to analyse the behaviour of the recommender systems in this domain. We propose the analysis of the list of recommended resources in three dimensions: the pedagogical utility, the similarity with the query and the diversity among the resources.

However, we should not forget that the strategies presented here are designed a learning environment. Therefore it is necessary to analyse the acceptance of the recommendation strategies by the users, as well as the satisfaction about the use of them. For this purpose we have carried out a survey with educators and students, who used the strategies during a certain time.

The chapter is organized as follows: Section 10.1 first detail the application of the strategies to an educational repository of programming LOs.

Section 10.2 presents the evaluation of our proposed strategy. This evaluation is two-fold: experimental analysis of the behaviour (Section 10.2.1) and the case study with real users (Section 10.2.2). Last section concludes the chapter.

10.1. Applying the approach to an educational repository of programming learning objects

The strategies described in this thesis have been successfully applied in a semi-formal learning scenario, where both educators and students use a repository of Computer Science (CS) LOs for computer programming in CS1-CS2 levels. This repository has been developed following a producer-consumer model (Downes, 2007) by a group of CS educators at the University Complutense de Madrid (Spain). The contents and the access facilities have evolved along the last five years from a set of solved examples organized by topic and free access to any resource using a topic browsing tool, to solved examples together with assignments organized in terms of a something rigid course (Gómez-Albarrán et al., 2011), to the current recommendation interface and the existence of solved examples, assignments and quiz questions. The types and granularity level of the LOs in the repository, together with the recommendation facilities that provide a user with resources matching her context and semantically similar to a query representing her needs, are a clear contribution in the field of LOs in CS learning (Dichev y Dicheva, 2012).

The sources involved in the prototypes are:

- The items: The dataset of LOs employed in the experiment consists of more than 500 LOs corresponding to exercises, explained examples and question test for learning Computer Programming. Each LO covers, on average, four concepts from our domain ontology.
- The domain ontology: The ontology is populated with 34 classes that represents concepts about basic computer programming and 28 instances that exemplifies these concepts in different programming languages. Concepts are organized in a taxonomy using the typical relation *is_a*. For instance, we have defined the *Iterative* concept that subsumes the concepts related to the control structures *count controlled loops*, *post condition check loops* and *condition controlled loops* (*for*, *do-while* and *while* loops are examples of these structures, respectively, in the C++ programming language).
- Contextual information: We have defined a precedence property in the ontology employed to represent a typical sequence or order in which

CS1 programming concepts are taught. For instance, the *Iterative* concept should be learned after the *Basic* concept (related to the usual predefined data types of programming languages), and before the *Arrays* concept. This property is related to the activity context. User context refers to the knowledge that the user achieves about the domain topics. It is modeled following a weighted overlay on the fine-grained conceptual structure defined in the ontology. More specifically, user context is represented by the mastery level achieved in lower-level concepts in the ontology and propagated over the domain concepts using the taxonomical representation. This mastery level is a rough estimate of the user knowledge in a particular concept. Many researchers recognize the benefits of this conventional ontology-based overlay user modeling (Sosnovsky y Dicheva, 2010).

10.2. Evaluation of the strategies

The utility of the described strategies must be supported by an evaluation of their performance according to the retrieved LOs and the user opinions. In this sense, next subsections summarize the methodology and the results achieved by our two-fold evaluation. A complete description of the evaluations can be found in (Ruiz-Iniesta et al., 2011a,b).

The first step could be to analyse the strategies behaviour according to the characteristics of recommendation list (similarity, pedagogical utility and diversity) regardless the order and size of the recommendation list. This analysis aims to show how the individual relevancies (similarity, pedagogical utility or diversity) of the recommendation list from the alpha parameter variations in the quality metrics used in the rating stage, regardless of the size of the set of LOs or the order.

On the one hand, we analyse the strategies performance modifying the parameters that affect their behaviour: α and k . The results of the analysis are the parameter values that make the strategy retrieve a balanced set of LOs according to the similarity with the query and their utility to improve user knowledge. This first evaluation will be carried out on the personalized knowledge-based recommendation strategy (*KBpersonalization*) and on the knowledge-based recommendation strategy that combines personalization and diversity (*KBdiversity*). On the other hand, the resulting values are employed to create a functional prototype that has been evaluated by educators and students. The evaluation will demonstrate whether both kinds of participants would embrace a tool that implements this strategy as a new way to interact with LOs repositories.

10.2.1. Experimental analysis of the behaviour

To perform these experiments we have developed four prototypes from the framework presented in Chapter 2, two related to *KBpersonalization* and two related to *KBdiversity*.

The prototypes related to *KBpersonalization* consist of the following classes from the framework:

- `FromFileQueryElicitationStage`.
- `ApproximateRetrievalStage`.
- `ReachableConceptsFilter`.
- The rating stage makes up two individual relevancies: `QuerySimilarity` and `PedagogicalUtility`.

- One of the prototypes makes up `WeightedMeanMetric`, C_{1p} :

$$C_{1p}(L, S, Q) = \alpha \cdot \text{Sim}(L, Q) + (1 - \alpha) \cdot \text{PU}(L, S) \quad \text{where } \alpha \in [0, 1] \quad (10.1)$$

- The second one makes up `HarmonicWeightedMeanMetric`, C_{2p} :

$$C_{2p}(L, S, Q) = \frac{1}{\frac{\alpha}{\text{Sim}(L, Q)} + \frac{(1-\alpha)}{\text{PU}(L, S)}} \quad \text{where } \alpha \in [0, 1] \quad (10.2)$$

- `TopKSelectionStage` with k half size of the set of LOs which reach this stage.

The prototypes developed to *KBdiversity* on the stages of query elicitation, retrieval and filtering are similar to the above. For the remaining stages have the following classes:

- The rating stage makes up the individual relevance: `QuerySimilarity`.
- The selection stage implements the diversity-conscious selection algorithm.

- One of the prototypes makes up `RelDWeightedMeanSelection`, C_{1d} :

$$C_{1d}(L, P, Q) = \alpha \cdot \text{Sim}(L, Q) + (1 - \alpha) \cdot \text{RelD}(L, P) \quad \text{where } \alpha \in [0, 1] \quad (10.3)$$

- The second one makes up `RelDHarmonicMeanSelection`, C_{2d} :

$$C_{2d}(L, P, Q) = \frac{1}{\frac{\alpha}{Sim(L, Q)} + \frac{(1-\alpha)}{RelD(L, P)}} \quad \text{where } \alpha \in [0, 1] \quad (10.4)$$

As far as the relative diversity metric *RelD* is concerned, we adopted the next metric:

$$RelD(L, P) = \begin{cases} 1 & \text{si } P = \{\} \\ \frac{\sum_{L_i \in P} (1 - Sim(L, L_i))}{|P|} & \text{in other case} \end{cases} \quad (10.5)$$

The dataset of LOs employed in the experiment consists of more than 500 LOs corresponding to exercises, explained examples and question test for learning Computer Programming. Each LO covers, on average, four concepts from our domain ontology. The ontology is populated with 37 concepts in the field of Computer Programming. An average each LO covers four concepts of the domain ontology. We have created a synthetic set of 30 heterogeneous student profiles. These profiles represent students who have explored approximately the 80% of the learning path in the ontology. The profiles are heterogeneous according to the average competence level of the students. For each student, we have employed 18 different queries. This way, we have performed 540 different recommendations using each quality metric. We repeated the recommendation process with different values of α —ranging from 0 to 1 in intervals of 0.1.

The analysis of the first strategy (*KBpersonalization*) shows that if obtain interesting levels of personalization while preserving the similarity with the query. Regarding the second strategy (*KBdiversity*) we have seen that it obtains interesting levels of diversity among the recommended LOs while preserving the similarity with the query. This analysis also shows that the second strategy lets introduce a light level of pedagogical utility.

The strategy performance has been analysed by studying the ranked lists of recommended LOs in two dimensions: the pedagogical utility of the LOs contained in the list and their similarity with the query for *KBpersonalization* and the diversity among the recommended LOs and their similarity with the query in the case of *KBdiversity* (Ruiz-Iniesta et al., 2011b).

Due to the importance of short recommendation lists, we consider crucial that the most relevant LOs appear in the upper positions of the recommendation. To evaluate this fact, we employ in our experiments the Normalized Discounted Cumulative Gain (NDCG), which has received an increasing adoption in the Information Retrieval field. This metric measures the usefulness or gain of a result list based on the relevance and the position of the retrieved documents and it compares the obtain gain with the ideal one.

For experimental purposes, we have modified the NDCG metric in order to analyse the LOs in the recommendation list with respect to their peda-

gical utility to the user U ($NDCG_{PU}$), their similarity with the query Q ($NDCG_{Sim}$) and the diversity among the LOs ($NDCG_{Div}$). We computed $NDCG_{PU}$, $NDCG_{Sim}$ and $NDCG_{Div}$ values for a list of size k using the following equations:

$$\begin{aligned} NDCG_{Sim}(L, Q) &= \frac{DCG_{Sim}(L, Q)}{IDCG_{Sim}(L', Q)} \\ &= \frac{Sim(L_1, Q) + \sum_{i=2}^k \frac{Sim(L_i, Q)}{\log_2 i}}{IDCG_{Sim}(L', Q)} \end{aligned} \quad (10.6)$$

$$\begin{aligned} NDCG_{PU}(L, S) &= \frac{DCG_{PU}(L, S)}{IDCG_{PU}(L', S)} \\ &= \frac{PU(L_1, S) + \sum_{i=2}^k \frac{PU(L_i, S)}{\log_2 i}}{IDCG_{PU}(L', S)} \end{aligned} \quad (10.7)$$

$$\begin{aligned} NDCG_{Div}(L) &= \frac{DCG_{Div}(L)}{IDCG_{Div}(L')} \\ &= \frac{RelD(L_1, \{\}) + \sum_{i=2}^k \frac{RelD(L_i, \{L_1, L_2, \dots, L_{i-1}\})}{\log_2 i}}{IDCG_{Div}(L')} \end{aligned} \quad (10.8)$$

where $IDCG_{PU}(L', S)$ and $IDCG_{Sim}(L', Q)$ are the DCG values of the list L' sorted by $PU(L, S)$ and $Sim(L, Q)$, respectively. The $DCG_{Div}(L)$ value is the relative diversity obtained in the selection stage. $IDCG_{Div}(L')$ value is the DCG of the list L' which is obtained by rearranging the elements of L applying the diversity preserving algorithm (Smyth y McClave, 2001), in which the quality metric is the relative diversity. The list L' , the same size as L , has as the first element that higher value of similarity to the query (previously calculated in the rating stage) and the following will be introduced to the diversity shown on the list which is built in each step of the algorithm.

Additionally, we would like to know which α -value would be the best in these experimental settings: the value of α that guarantees the highest compromise between the achievement of the long-term and short-term learning goals. In other words, the best α -value is the one where the loss of pedagogical utility relative to the optimal long-term strategy ($\alpha = 0$) and the loss of similarity relative to the optimal short-term strategy ($\alpha = 1$) are minimal. For this purpose, we have defined a function that represents this compromise:

$$\begin{aligned} Comp_{\alpha, PU-Sim}(L, S, Q) &= \\ &= (NDCG_{PU}(L, S)_{\alpha=0} - NDCG_{PU}(L, S)_{\alpha}) + \\ &+ (NDCG_{Sim}(L, Q)_{\alpha=1} - NDCG_{Sim}(L, Q)_{\alpha}) \end{aligned} \quad (10.9)$$

$$\begin{aligned}
Comp_{\alpha, Div-Sim}(L, Q) = & \\
& (NDCG_{Div}(L)_{\alpha=0} - NDCG_{Div}(L)_{\alpha}) + \\
& + (NDCG_{Sim}(L, Q)_{\alpha=1} - NDCG_{Sim}(L, Q)_{\alpha})
\end{aligned}
\tag{10.10}$$

To run the computational evaluation we have modelled a set of 30 heterogeneous student profiles. These profiles represent students who have explored approximately the 80% of the learning path in the ontology. For each student, we employed 18 different queries with 1 to 4 concepts each. This way, we performed 540 different recommendations. We repeated the recommendation process with different values of α —ranging from 0 to 1 in intervals of 0.1— and k —5, 10 and 20 items per recommendation. The $NDCG_{PU}$, $NDCG_{Sim}$ and $NDCG_{Div}$ values over all recommended lists was averaged to yield a single quantitative metric for each pair of α and k values.

The results for both recommendation strategies show that they obtain high levels of similarity gain. In the retrieval stage we use an approximate retrieval. This method establishes that a LO is retrieved if it covers at least one of the concepts of the query (regardless of the other concepts that it covers). Although we use this kind of retrieval, the results of $NDCG_{Sim}$ show that the similarity with the query is preserved in the recommended list. We consider that this is a good behaviour. As far as the tendency of $NDCG_{PU}$ is concerned, it differs depending on α and on the size of the list. However, any of the results obtained remains in high values of $NDCG_{PU}$. Finally, as far as the $NDCG_{Div}$ is concerned the results show that this relevance is not affected by the different parameter values.

In general, we can stress that the knowledge-based strategies obtains high values for PU , so the strategies propose recommendations that satisfy the long-term learning goals. Additionally, if we analyse the behaviour of the similarity with the query, in *KBpersonalization*, we can see that the recommendation strategy always ensures that the proposed LOs meet the short-term goals. On the other hand, the results also show that *KBdiversity* obtains high levels of diversity.

10.2.2. Evaluation with users

Once the experimental analysis was completed, we developed three functional prototype and started an evaluation with users. Each prototype corresponds to one of the proposed strategies.

The main goal of our evaluation with users is to determine the impact that the use of the context-aware recommendation strategy described has on the learning-teaching process from the perspective of students and educators. This evaluation lets us know the weaknesses of the recommendation strategy and improve it by developing alternatives to the ones described here.

This goal can be broken down into three sub-goals based on the ones proposed in the Unified Theory of User Acceptance of Information Technology (Venkatesh et al., 2003):

- Performance expectancy: the degree to which students and instructors believe that using the recommendation strategy will enhance the learning process.
- Effort expectancy: the degree of ease associated with the use of the recommendation strategy.
- Behavioural intention to use the system: the degree to which students and instructors would use the recommendation strategy in other LOs repositories and learning scenarios.

We have decided to employ a goal-oriented evaluation based on the Goal-Question-Metric (GQM) method (Basili y Rombach, 1988). This method points out that, in order to improve a process, we have to define measurement goals, which will be refined into questions, and, consecutively, into metrics which will supply all the necessary information for answering those questions. The analysis and interpretation of the answers helps us to assess whether the goals were attained (Solingen y Berghout, 1999).

Each sub-goal was refined into several questions: 3 questions related to performance expectancy, 5 questions related to effort expectancy and 2 questions related to intention of use. These 10 questions were assembled in a survey whose answers were defined as ratings on a five-point Likert Scale. Additionally, users could provide free-text comments to explain the ratings provided for each question or suggest improvements in the recommender.

We developed three LO recommender prototypes for evaluation purposes. Each of one supports formal computer programming learning with a repository of LOs that are used for the blended learning of CS1 in our university. The system works in Windows/Mac/Linux machines with Java support and Internet connection. The system follows a client/server model. The server contains the dataset of LOs, the domain ontology and the contextual data. The client provides a user interface that is responsible for building the query, generating the recommendation and communicating with the server through the network to provide access to the LOs recommended.

The first prototype, *KBpersonalization*, according to the results of the analysis described in the previous section, we configured the system, setting Equation 8.1 as quality metric in the rating stage and $\alpha = 0.2$ –the parameter in Equation (8.1)– because it keeps the best compromise between similarity and the utility assigned to the student’s context information. The number of LOs recommended in response to a query was fixed at $k = 7$, which lies in the interval $[5, 10]$ and provides reasonable results, as detailed in the previous section.

The second prototype, *KBdiversity*, has the Equation 8.6 as quality metric in the selection stage. The parameter has been set to 0.1, because it keeps the best compromise between similarity and the diversity. The number of LOs recommended in response to a query was fixed at $k = 7$.

The third prototype, *KBnavigation*, uses Equation 8.1 as quality metric with $\alpha = 0.2$. The selection stage shows at least one member of each group in which the repository has been divided being the minimum number of recommended LOs $k = 7$.

The survey took place at the Computer Science School at Complutense University of Madrid during 2011-12 and 2012-13 academic years. Participants (students and educators) attended a guided session about how the tool works. Afterwards, they freely interacted with the tool in a 60-minute session, establishing a user profile and creating different queries in order to retrieve LOs related to different CS1-CS2 learning concepts. Educators completed a questionnaire after that session. Students could continue interacting with the tool during 90 days more and then they also completed the questionnaire. The questionnaire contained questions, which could be answered using a five-point Likert Scale (1. Strongly disagree–5. Strongly agree). Additionally, we asked the participants about two possible improvements for the recommendation tool: providing explanations about the recommended LOs, and providing support for free-text queries in order to ease the search of desired concepts in the taxonomy. Finally, participants could provide free-text comments that helped us to understand their ratings. 14 educators and 171 students participated in the survey. Educators interacted with the three prototypes. Students were divided into groups of 51, 70 and 50 member, and each group used one of the three prototypes.

Figures 5.14, 5.15, 5.16 in the Spanish version, summarize the responses concerning the three sub-goals in the online survey for *KBpersonalization*. Figures 5.17, 5.18, 5.19 in the Spanish version, summarize the responses of *KBdiversity*. Finally, Figures 5.20, 5.21, 5.22 in the Spanish version, summarize the responses concerning *KBnavigation*.

The surveys show that the tool helps in their learning-teaching activities because it helps to find resources faster and the retrieved resources fit their interest and knowledge (or that of their audience). In general, the tool showed a lightly steep learning curve and users showed interest to also use the tool with LOs repositories in different domains.

As Figures show, there is a high agreement about the benefit of using the recommendation tools. Most of the students and educators agree that the recommendation tools help to find interesting LOs according to a given user profile and similar to the posed query, but there are differences between students and educators regarding to how fast they can find useful LOs in *KBpersonalization*. Most educators believe that the tools will be helpful for both students and educators but less students than educators agreed this

statement. Further comments revealed that some students had networking problems and they spent long time getting used to the taxonomy (in *KBpersonalization* and *KBdiversity*).

Despite the conclusions extracted above, the results of the questions related to effort expectancy reveal that the recommendation tools are, in general, easy to use. Most of the participants consider that they can generate queries easily (for *KBpersonalization* and *KBdiversity*), but these ratings are a little bit lower when asking specifically about the concept hierarchy understanding. The results provided by the recommendation tools were clear and understandable for most of the participants and it is worth noting the slight disagreement between students and instructors about the number of LOs recommended.

Regarding the intention of using the tools, most of the educators and students show an interest to use the recommendation tools in other disciplines and would encourage other instructors or students to use it

Further student comments revealed that over 50% of them consider it interesting to include explanations that provide information about why an LO has been recommended. This result suggest to us what the next steps could be to improve the strategy.

10.2.3. Conclusions

In this chapter we have presented the evaluation of the knowledge-based recommendation strategies. The knowledge-based recommendation strategies proposed have gone through two types of evaluation. The first one –the experimental evaluation– analysed the behaviour of the recommendation strategy according to the similarity to the query and its pedagogical utility. This strategy relies on a parameterized quality metric computed as the weighted sum of the similarity between the LO and the query and the pedagogical utility of the LO to the current student.

The other evaluation concerns student satisfaction. We analysed performance expectancy, effort expectancy and intention of use of three recommender prototypes that follow the approaches described. According to the students' point of view –and further comments provided by the students–, the evaluation results show that students do believe that the recommendation tools are a good tool for supplementing their learning activities. They would find the use of this kind of tool in other courses interesting and they consider that the learning curve of the system is low. The survey pointed out the need for one concrete improvements: the inclusion of explanations. This aspect will be considered in our short-term future work.

Chapter 11

Conclusions and future work

In this chapter we review the main contributions of this thesis. We also summarize future research lines to improve our approach.

11.1. Conclusions

This thesis proposes three knowledge-based recommendation strategies that supports a personalized access to the LOs contained in a repository, according to the contextual information about the student and about the activity. The recommendation strategies rely on LOs semantic description using LOM metadata and indexation in an ontology of domain concepts.

The first strategy allows to incorporate high levels of personalization in the recommendation process. The second strategy promotes the diversity among the recommended resources avoiding the overspecialization problem. The last strategy explores a user interaction model based in a navigation by proposing model.

We have also designed a framework that allows the rapid prototyping of knowledge-based recommender for educational domain. This framework was designed based on the common characteristics presented in all three strategies proposed, obtaining a generic recommendation process and completely configurable.

Finally, these three strategies have also suffered an evaluation from a computational point of view and from the direct application in a real situation of teaching and learning with educators and students.

Each strategy alleviates to one of the weaknesses identified in the related work and attempts to remedy the needs that users of this kind of repositories missed and the weaknesses of previous work analysed. The main aspects of these strategies are:

1. They profit from a description of the educational resources that leverage the potential of metadata standards (concretely IEEE LOM) and

enriches them through an ontology-based semantic indexing. In this sense, we agree with other authors on the use of semantic descriptions in order to facilitate the discovery of digital educational contents. Semantically enriched descriptions of the resources and an ontology gathering both the domain vocabulary and relations among vocabulary terms are crucial to locate resources that satisfy the user needs.

2. They use contextual information about the user. In the recent years the contextual information has achieved a distinguished role in the recommendation domain. Our work joins this new trend of research efforts to find educational contents matching a specific context, but makes the most of other kind of context: the learner's knowledge.
3. They use contextual information about the activity. This knowledge is related to the inclusion of learning paths. A learning path reflects a traditional sequence or order in which concepts are taught or learned in the corresponding field.
4. They provide LOs that adapts to the student goals and lets introduce diversity in the recommendation results. One of the strategies explores an improved knowledge-based recommendation strategy that promotes the inclusion of personalization. As far as the diversity aspect is concerned, the second strategy avoids that LOs that are highly correlated with the student goals, and highly correlated among themselves, are the only ones recommended. Introducing diversity in the recommended LOs is also crucial for making the most of the recommendation session.
5. They explore two models of user interaction. The first one is a reactive model that requires the students had to pose a query that represents her short-term learning goals in terms of the topics she is interested in. The second one is a navigation-by-proposing interaction model, a simple conversational process that avoids posing direct questions and carries very little feedback overhead from the students' perspective.

In summary, the main contributions of this thesis are:

- We have presented a review of the current state of the research on recommender systems. This review allowed us to establish what are the main features of these systems, their benefits and drawbacks.
- We have carried out an analysis on the transfer of recommender systems to the learning field. This analysis has allowed us to understand what are the main features in these environments.
- We have detected three lines of improvement in the transfer of recommender systems to the learning field:

- Including a high level of personalization making use of contextual information and domain knowledge.
 - Design of a selection strategy that tackles the problem of overspecialization and promotes diversity.
 - Providing a recommender-user interaction strategy through navigation by proposing.
- We have presented a first recommendation strategy that works on the LOs contained in digital repositories and makes use of contextual information about the activity and the user (Ruiz-Iniesta et al., 2009a,c, 2010). Our activity-related contextual information refers to the inclusion of successful learning paths and strategies that could provide guiding principles for recommendation. Our user context is based on the learning goals, defined by a domain expert, that the learner achieves during the learning process. This context-aware recommendation strategy gives way to a new and successful personalized access approach to LO repositories.
 - We have presented a second knowledge-based recommendation strategy that combines the inclusion of personalization and diversity (Ruiz-Iniesta et al., 2011a). Our knowledge-based strategy constitutes a quite novel approach in accessing educational repositories. As far as we know, there are no other works on recommendation in e-learning that face the overspecialization problem or lack of diversity.
 - We have finally presented a proactive recommendation strategy that proposes LOs according to the contextual information about the user and about the activity. It exploits a type of conversational technique that imposes little feedback overhead.
 - We have detailed the knowledge sources employed by our recommendation approaches: the LOs, the domain ontology and the contextual information.
 - We have developed a framework for rapid prototyping of knowledge-based recommender systems of learning objects (Ruiz-Iniesta et al., 2011c, 2012b). In order to design the framework, we have analysed the recommendation process. This analysis identified 5 stages in the process, each one was considered as a shaft of variability in the recommendation process. The framework was designed so that the developer can easily create recommenders that use alternative strategies, implemented for each stage. It was also considered that the framework can easily be increased by implementing new strategies.
 - We have applied the the recommendation strategies in a LOs repository of Computer Programming LOs employed in CS1 courses in the Com-

puter Science School at the Complutense University of Madrid. This application has led to: (1) to develop an OWL ontology that contains the domain concepts and learning paths, (2) to adapt the resources to LOM, and (3) to develop a prototype (using the framework developed) for each of the proposed strategies that could be used by real students.

- The recommendation strategies have been tested in a repository of CS1-CS2 programming LOs. They have suffered a two-fold evaluation. On the one hand, we made an experimental evaluation to analyse their performance according to the similarity, diversity and pedagogical utility (Ruiz-Iniesta et al., 2011a,b). The analysis suggested that we could obtain interesting levels of pedagogical utility while preserving similarity with the query. Performance results helped us to tune the strategy parameters in order to balance the resulting list of LOs according to the relevancies. Afterwards, a survey with users (students and educators) analysed the impact that the recommendation strategies introduce in the use of an LOs repository and in the learning-teaching process, in terms of the performance expectancy, effort expectancy and behavioural intention to use the recommender in the future. In general, the users agree in that the tool showed a lightly steep learning curve and users showed interest to also use the tool with LOs repositories in different domains.

11.2. Future work

Next we describe some future work:

- The surveys provided by students pointed out the user interest in a future improvement: the support for free-text search through the taxonomy of concepts. Some users initially had difficulties when posing the query in terms of the concept hierarchy and they considered interesting a free-text search facility when we explicitly asked them about its inclusion in further prototypes. Users also considered interesting obtaining explanations about the reasons why an LOs was included in the recommendation. Both aspects will be considered in our short-term future work.
- The survey also pointed out the need for the inclusion of explanations. The explanations are related to increase user confidence in the system. So, it would be interesting that each recommended resource lead an associated explanation of why has been recommended.
- We have used the simplest form of feedback which involves the user in indicating a simple preference for one recommended item over another.

This approach carries very little feedback overhead, from the students perspective, but it provides a limited guide in the recommendation process. So we will refine the third strategy by inviting students to offer more accurate feedback (for example, by showing the aspects reflected in the LO description and requesting students to mark the ones that capture her interest). This could reduced the conversational cycle.

- As far as the framework is concerned, we plan to increase the concrete classes of the framework with the improvements mentioned above. We are also exploring how to extend the architecture of the framework in order to include hybrid recommendation strategies. We are also considering the inclusion of new classes that allow the automation to the evaluation tasks of the framework. We also plan to include the next improvements in the recommendation process:
 - We plan to include, the information about the navigation history recorded by the student and add this information to the contextual information. This information will be included in order to mark the recommended LOs that the student has already explored. We plan to use this information in a new filter that discards those LOs that the student has marked as not relevant.
 - Regarding to the aggregation functions employed, we plan to include new aggregation quality metrics in order to combine the individual utilities, such as the ones employed in social group recommendations (Masthoff, 2004).
 - It desirable to include new utility functions that take into account other features of the LOs, for example the kind of the resources or the difficulty level.
 - We also plan to include new classes, in the selection stage, that let include diversity using alternative algorithms than the actually employed.
- Finally, we propose to study the use of these tools in other disciplines. For the transfer of these strategies would be necessary to analyze what are the knowledge needs and how to adapt the knowledge sources.

Bibliografía

- ADOMAVICIUS, G., MANOUSELIS, N. y KWON, Y. Multi-criteria recommender systems. En *Recommender Systems Handbook*, páginas 769–803. Springer, 2011a. ISBN 978-0-387-85820-3.
- ADOMAVICIUS, G., MOBASHER, B., RICCI, F. y TUZHILIN, A. Context-Aware recommender systems. *AI Magazine*, vol. 32(3), páginas 67–80, 2011b. ISSN 0738-4602.
- ADOMAVICIUS, G., SANKARANARAYANAN, R., SEN, S. y TUZHILIN, A. Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Transactions on Information Systems*, vol. 23(1), páginas 103–145, 2005. ISSN 1046-8188.
- ADOMAVICIUS, G. y TUZHILIN, A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, vol. 17(6), páginas 734–749, 2005. ISSN 1041-4347.
- AHN, J.-W., BRUSILOVSKY, P., GRADY, J., HE, D. y SYN, S. Y. Open user profiles for adaptive news systems: help or harm? En *Proceedings of the 16th International Conference on World Wide Web*, páginas 11–20. ACM, 2007. ISBN 978-1-59593-654-7.
- ALUR, D., MALKS, D., CRUPI, J., BOOCH, G. y FOWLER, M. *Core J2EE Patterns (Core Design Series): Best Practices and Design Strategies*. Prentice Hall, 2003. ISBN 978-0131422469.
- ANAND, S. S. y MOBASHER, B. Contextual recommendation. En *From Web to Social Web: Discovering and Deploying User and Content Profiles*, páginas 142–160. Springer, 2007. ISBN 978-3-540-74950-9.
- ATKINSON, R. y SHIFFRIN, R. Human memory: A proposed system and its control processes. *The Psychology of Learning and Motivation*, vol. 2(1), páginas 89–195, 1968. ISSN 0079-7421.

- AVANCINI, H. y STRACCIA, U. User recommendation for collaborative and personalised digital archives. *International Journal of Web Based Communities*, vol. 1(2), páginas 163–175, 2005. ISSN 1477-8394.
- BALABANOVIC, M. An adaptive web page recommendation service. En *Proceedings of the 1st International Conference on Autonomous Agents*, páginas 378–385. ACM, 1997. ISBN 0-89791-877-0.
- BALABANOVIC, M. y SHOHAM, Y. Fab: content-based, collaborative recommendation. *Communications of ACM*, vol. 40(3), páginas 66–72, 1997. ISSN 0001-0782.
- BASILI, V. R. y ROMBACH, H. D. The TAME project: towards improvement-oriented software environments. *IEEE Transactions on Software Engineering*, vol. 14(6), páginas 758–773, 1988. ISSN 0098-5589.
- BASU, C., HIRSH, H. y COHEN, W. Recommendation as classification: Using social and content-based information in recommendation. En *Proceedings of the 15th National Conference on Artificial Intelligence*, páginas 714–720. AAAI Press, 1998. ISBN 0-262-51098-7.
- BELIAKOV, G., CALVO, T. y JAMES, S. Aggregation of preferences in recommender systems. En *Recommender Systems Handbook*, páginas 705–734. Springer, 2011. ISBN 978-0-387-85819-7.
- BELL, R. M. y KOREN, Y. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. En *Proceedings of the 7th IEEE International Conference on Data Mining*, páginas 43–52. IEEE Computer Society, 2007. ISBN 0-7695-3018-4.
- BILLSUS, D. y PAZZANI, M. J. A hybrid user model for news story classification. En *Proceedings of the 7th International Conference on User Modeling*, páginas 99–108. Springer, 1999. ISBN 3-211-83151-7.
- BIN GHAUTH, K. y ABDULLAH, N. Building an e-learning recommender system using vector space model and good learners average rating. En *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies*, páginas 194–196. IEEE Computer Society, 2009. ISBN 978-0-7695-3711-5.
- BLEI, D. M., NG, A. Y. y JORDAN, M. I. Latent dirichlet allocation. *The Journal of Machine Learning Research*, vol. 3(mar), páginas 993–1022, 2003. ISSN 1532-4435.
- BOBROW, D. Gus, a frame driven dialog system. En *Readings in natural language processing*, páginas 595–604. Morgan Kaufmann Publishers Inc., 1986. ISBN 0-934613-11-7.

- BOLLACKER, K. D., LAWRENCE, S. y GILES, C. L. CiteSeer: an autonomous web agent for automatic retrieval and identification of interesting publications. En *Proceedings of the 2nd International Conference on Autonomous Agents*, páginas 116–123. ACM, 1998. ISBN 0-89791-983-1.
- BRADLEY, K., RAFTER, R. y SMYTH, B. Case-Based user profiling for content personalisation. En *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, páginas 62–72. Springer, 2000. ISBN 3-540-67910-3.
- BREESE, J. S., HECKERMAN, D. y KADIE, C. Empirical analysis of predictive algorithms for collaborative filtering. En *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, páginas 43–52. Morgan Kaufmann Publishers Inc., 1998. ISBN 1-55860-555-X.
- BRIDGE, D. Towards conversational recommender systems: A dialogue grammar approach. En *Proceedings of the Workshop in Mixed-Initiative Case-Based Reasoning at the 6th European Conference in Case-Based Reasoning*, páginas 9–22. The Robert Gordon University, 2002.
- BULANDER, R., DECKER, M., SCHIEFER, G. y KOLMEL, B. Comparison of different approaches for mobile advertising. En *Proceedings of the 2nd IEEE International Workshop on Mobile Commerce and Services*, páginas 174–182. IEEE Computer Society, 2005. ISBN 0-7695-2391-9.
- BURKE, R. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, vol. 12(4), páginas 331–370, 2002a. ISSN 0924-1868.
- BURKE, R. Interactive critiquing for catalog navigation in E-Commerce. *Artificial Intelligence Review*, vol. 18(3-4), páginas 245–267, 2002b. ISSN 0269-2821.
- BURKE, R. Hybrid web recommender systems. En *The Adaptive Web*, páginas 377–408. Springer, 2007. ISBN 978-3-540-72078-2.
- BURKE, R. D. Knowledge-based recommender systems. *Encyclopedia of Library and Information Systems*, vol. 69(Sup. 32), 2000.
- BURKE, R. D., HAMMOND, K. J. y YOUNG, B. C. Knowledge-based navigation of complex information spaces. En *Proceedings of the 13th National Conference on Artificial Intelligence*, páginas 462–468. AAAI Press, 1996. ISBN 0-262-51091-X.
- BURKE, R. D., HAMMOND, K. J. y YOUNG, B. C. The FindMe approach to assisted browsing. *IEEE Expert: Intelligent Systems and Their Applications*, vol. 12(4), páginas 32–40, 1997. ISSN 0885-9000.

- CANTADOR, I., BELLOGÍN, A. y VALLET, D. Content-based recommendation in social tagging systems. En *Proceedings of the 4th ACM Conference on Recommender Systems*, páginas 237–240. ACM, 2010. ISBN 978-1-60558-906-0.
- CARBONELL, J. y GOLDSTEIN, J. The use of MMR, diversity-based reranking for reordering documents and producing summaries. En *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, páginas 335–336. ACM, 1998. ISBN 1-58113-015-5.
- CASTAGNOS, S., JONES, N. y PU, P. Eye-tracking product recommenders usage. En *Proceedings of the 4th ACM Conference on Recommender Systems*, páginas 29–36. ACM, 2010. ISBN 978-1-60558-906-0.
- CHEN, L. y PU, P. Preference-Based organization interfaces: Aiding user critiques in recommender systems. En *Proceedings of the 11th International Conference on User Modeling*, páginas 77–86. Springer, 2007. ISBN 978-3-540-73077-4.
- COYLE, L., CUNNINGHAM, P. y HAYES, C. A Case-Based personal travel assistant for elaborating user requirements and assessing offers. En *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, páginas 505–518. Springer, 2002. ISBN 3-540-44109-3.
- DESHPANDE, M. y KARYPIS, G. Item-based top- N recommendation algorithms. *ACM Transactions on Information Systems*, vol. 22(1), páginas 143–177, 2004. ISSN 1046-8188.
- DESROSIERS, C. y KARYPIS, G. A comprehensive survey of neighborhood-based recommendation methods. En *Recommender Systems Handbook*, páginas 107–144. Springer, 2011. ISBN 978-0-387-85819-7.
- DICHEV, C. y DICHEVA, D. Open educational resources in computer science teaching. En *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, páginas 619–624. ACM, 2012. ISBN 978-1-4503-1098-7.
- DIEDERICH, J. y IOFCIU, T. Finding communities of practice from user profiles based on folksonomies. En *Proceedings of the 1st International Workshop on Building Technology Enhanced Learning Solutions for Communities of Practice*, páginas 288–297. 2006. ISSN 1613-0073.
- DOWNES, S. Models for sustainable open educational resources. *Interdisciplinary Journal of Knowledge and Learning Objects*, vol. 3, páginas 29–44, 2007. ISSN 1552-2237.

- DOYLE, M. y CUNNINGHAM, P. A dynamic approach to reducing dialog in on-line decision guides. En *Advances in Case-Based Reasoning*, páginas 49–60. Springer, 2000. ISBN 978-3-540-67933-2.
- DRACHSLER, H., HUMMEL, H., VAN DEN BERG, B., ESHUIS, J., WATERINK, W., NADOLSKI, R., BERLANGA, A., BOERS, N. y KOPER, R. Effects of the ISIS recommender system for navigation support in self-organised learning networks. *Educational Technology & Society*, vol. 12(3), páginas 115–126, 2009. ISSN 1436-4522.
- DRACHSLER, H., HUMMEL, H. y KOPER, R. Recommendations for learners are different: Applying memory-based recommender system techniques to lifelong learning. En *Proceedings of the 1st Workshop on Social Information Retrieval for Technology-Enhanced Learning & Exchange*, páginas 18–26. CEUR-WS, 2007. ISSN 1613-0073.
- DRACHSLER, H., HUMMEL, H. G. y KOPER, R. Personal recommender systems for learners in lifelong learning networks; the requirements, techniques and model. *International Journal of Learning Technology*, vol. 3(4), páginas 404–423, 2008. ISSN 1477-8386.
- DRON, J., MITCHELL, R., SIVITER, P. y BOYNE, C. CoFIND an experiment in n-dimensional collaborative filtering. *Journal of Network and Computer Applications*, vol. 23(2), páginas 131–142, 2000a. ISSN 1084-8045.
- DRON, J., SIVITER, P., BOYNE, C. y MITCHELL, R. CoFIND: steps towards a self-organising learning environment. En *Proceedings of WebNet World Conference on the WWW and Internet*, vol. 2000, páginas 146–151. AACE, 2000b. ISBN 1-880094-40-1.
- DUVAL, E., FORTE, E., CARDINAELS, K., VERHOEVEN, B., VAN DURM, R., HENDRIKX, K., FORTE, M. W., EBEL, N., MACOWICZ, M., WARKENTYNE, K. y HAENNI, F. The ARIADNE knowledge pool system. *Communications of the ACM*, vol. 44(5), páginas 72–78, 2001. ISSN 0001-0782.
- EKSTRAND, M. D., LUDWIG, M., KOLB, J. y RIEDL, J. T. LensKit: a modular recommender framework. En *Proceedings of the 5th ACM Conference on Recommender systems*, páginas 349–350. ACM, 2011. ISBN 978-1-4503-0683-6.
- FAYAD, M. E., SCHMIDT, D. C. y JOHNSON, R. E. *Building application frameworks: object-oriented foundations of framework design*. John Wiley & Sons, Inc., 1999. ISBN 0-471-24875-4.
- FELFERNIG, A. y BURKE, R. Constraint-based recommender systems: technologies and research issues. En *Proceedings of the 10th International*

- Conference on Electronic Commerce*, páginas 3:1–3:10. ACM, 2008. ISBN 978-1-60558-075-3.
- FELFERNIG, A., ISAK, K., SZABO, K. y ZACHAR, P. The VITA financial services sales support environment. En *Proceedings of the 19th National Conference on Innovative Applications of Artificial Intelligence*, páginas 1692–1699. AAAI Press, 2007. ISBN 978-1-57735-323-2.
- FERNÁNDEZ-LÓPEZ, M., GÓMEZ-PÉREZ, A. y JURISTO, N. METHONTOLOGY: from ontological art towards ontological engineering. En *Proceedings of the AAAI Spring Symposium on Ontological Engineering*, páginas 33–40. AAAI, 1997. ISBN 978-1-57735-042-2.
- FLING, B. *Mobile Design and Development: Practical Concepts and Techniques for Creating Mobile Sites and Web Apps*. O'Reilly Media, Inc., 2009. ISBN 978-0-596-15544-5.
- FORBES, P. y ZHU, M. Content-boosted matrix factorization for recommender systems: experiments with recipe recommendation. En *Proceedings of the 5th ACM Conference on Recommender Systems*, páginas 261–264. ACM, 2011. ISBN 978-1-4503-0683-6.
- GAMMA, E., HELM, R., JOHNSON, R. E. y VLISSIDES, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN 0-201-63361-2.
- GANTNER, Z., RENDLE, S., FREUDENTHALER, C. y SCHMIDT-THIEME, L. MyMediaLite: a free recommender system library. En *Proceedings of the 5th ACM Conference on Recommender systems*, páginas 305–308. ACM, 2011. ISBN 978-1-4503-0683-6.
- DE GEMMIS, M., LOPS, P., SEMERARO, G. y BASILE, P. Integrating tags in a semantic content-based recommender. En *Proceedings of the 2008 ACM Conference on Recommender Systems*, páginas 163–170. ACM, 2008. ISBN 978-1-60558-093-7.
- GHAUTH, K. I. B. y ABDULLAH, N. A. The effect of incorporating good learners' ratings in e-learning content-based recommender system. *Educational Technology & Society*, vol. 14(2), páginas 248–257, 2011. ISSN 1436-4522.
- GOLDBERG, D., NICHOLS, D., OKI, B. M. y TERRY, D. Using collaborative filtering to weave an information tapestry. *Communications of ACM*, vol. 35(12), páginas 61–70, 1992. ISSN 0001-0782.
- GÓMEZ-ALBARRÁN, M., GÓMEZ-MARTÍN, M. A., DÍAZ-ESTEBAN, A., HERNÁNDEZ-YAÑEZ, L. y RUIZ-INIESTA, A. Experiencias con repositorios institucionales de recursos educativos: Del libre acceso a las secuencias

- guiadas de aprendizaje. *IEEE-RITA*, vol. 6(2), páginas 80–86, 2011. ISSN 1932-8540.
- GÓMEZ-ALBARRÁN, M. y JIMÉNEZ-DÍAZ, G. Recommendation and students authoring in repositories of learning objects: A case-based reasoning approach. *International Journal of Emerging Technologies in Learning*, vol. 4, páginas 35–40, 2009. ISSN 1863-0383.
- GONZÁLEZ-CALERO, P., DÍAZ-AGUDO, B. y GÓMEZ-ALBARRÁN, M. Applying DLs for retrieval in case-based reasoning. En *Proceedings of the International Workshop on Description Logics*, páginas 51–55. CEUR-WS, 1999. ISSN 1613-0073.
- GRUBER, T. R. A translation approach to portable ontology specifications. *Knowledge Acquisition*, vol. 5(2), páginas 199–220, 1993. ISSN 1042-8143.
- GUNAWARDANA, A. y MEEK, C. A unified approach to building hybrid recommender systems. En *Proceedings of the 3rd ACM Conference on Recommender Systems*, páginas 117–124. ACM, 2009. ISBN 978-1-60558-435-5.
- HANNON, J., BENNETT, M. y SMYTH, B. Recommending twitter users to follow using content and collaborative filtering approaches. En *Proceedings of the 4th ACM Conference on Recommender Systems*, páginas 199–206. ACM, 2010. ISBN 978-1-60558-906-0.
- HERLOCKER, J. y KONSTAN, J. Content-independent task-focused recommendation. *IEEE Internet Computing*, vol. 5(6), páginas 40–47, 2001. ISSN 1089-7801.
- HERLOCKER, J., KONSTAN, J. A. y RIEDL, J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Information Retrieval*, vol. 5(4), páginas 287–310, 2002. ISSN 1386-4564.
- HERLOCKER, J. L., KONSTAN, J. A., BORCHERS, A. y RIEDL, J. An algorithmic framework for performing collaborative filtering. En *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, páginas 230–237. ACM, 1999. ISBN 1-58113-096-1.
- HERLOCKER, J. L., KONSTAN, J. A., TERVEEN, L. G. y RIEDL, J. T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, vol. 22(1), páginas 5–53, 2004. ISSN 1046-8188.
- HOFMANN, T. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, vol. 22(1), páginas 89–115, 2004. ISSN 1046-8188.

- HUMMEL, H. G. K., BERG, B. V. D., BERLANGA, A. J., DRACHSLER, H., JANSSEN, J., NADOLSKI, R. y KOPER, R. Combining social-based and information-based approaches for personalised recommendation on sequencing learning activities. *International Journal of Learning Technology*, vol. 3(2), páginas 152–168, 2007. ISSN 1477-8386.
- IEEE. Draft standard for Learning Object Metadata. 15 July 2002.
- IMS GLOBAL LEARNING CONSORTIUM, Y. . . , EDITOR = KEVIN RILEY (IMS), MARK MCKELL (IMS). IMS digital repositories interoperability - core functions information model. Informe técnico, 2003
- JAMALI, M. y ESTER, M. A matrix factorization technique with trust propagation for recommendation in social networks. En *Proceedings of the 4th ACM Conference on Recommender Systems*, páginas 135–142. ACM, 2010. ISBN 978-1-60558-906-0.
- JANNACH, D. ADVISOR SUITE - A knowledge-based sales advisory-system. En *Proceedings of the 16th European Conference on Artificial Intelligence*, páginas 720–724. IOS Press, 2004. ISBN 1-58603-452-9.
- JANNACH, D., ZANKER, M., FELFERNIG, A. y FRIEDRICH, G. *Recommender Systems: An Introduction*. Cambridge University Press, 2010. ISBN 9780521493369.
- JEREMIC, Z., JOVANOVIĆ, J. y GASEVIC, D. Semantic web technologies for the integration of learning tools and context-aware educational services. En *Proceedings of the 8th International Semantic Web Conference*, páginas 860–875. Springer, 2009. ISBN 978-3-642-04929-3.
- JÄRVELIN, K. y KEKÄLÄINEN, J. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, vol. 20(4), páginas 422–446, 2002. ISSN 1046-8188. ACM ID: 582418.
- KIM, S. y KWON, J. Effective context-aware recommendation on the semantic web. *International Journal of Computer Science and Network Security*, vol. 7(8), páginas 154–159, 2007. ISSN 1738-7906.
- KIM, Y., OK, S. y WOO, Y. A case-based recommender system using implicit rating techniques. En *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, páginas 522–526. Springer, 2002. ISBN 3-540-43737-1.
- KONSTAN, J. A., MILLER, B. N., MALTZ, D., HERLOCKER, J. L., GORDON, L. R. y RIEDL, J. GroupLens: applying collaborative filtering to usenet news. *Communications of ACM*, vol. 40(3), páginas 77–87, 1997. ISSN 0001-0782.

- KOREN, Y. Factorization meets the neighborhood: a multifaceted collaborative filtering model. En *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining*, páginas 426–434. ACM, 2008. ISBN 978-1-60558-193-4.
- KOREN, Y. y BELL, R. Advances in collaborative filtering. En *Recommender Systems Handbook*, páginas 145–186. Springer, 2011. ISBN 978-0-387-85820-3.
- KOREN, Y., BELL, R. y VOLINSKY, C. Matrix factorization techniques for recommender systems. *Computer*, vol. 42(8), páginas 30–37, 2009. ISSN 0018-9162.
- KOREN, Y. y SILL, J. OrdRec: an ordinal model for predicting personalized item rating distributions. En *Proceedings of the 5th ACM Conference on Recommender Systems*, páginas 117–124. ACM, 2011. ISBN 978-1-4503-0683-6.
- LALIWALA, Z., SORATHIA, V. y CHAUDHARY, S. Semantic and rule based event-driven services-oriented agricultural recommendation system. En *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems Workshops*, páginas 24–24. IEEE Computer Society, 2006. ISBN 0-7695-2541-5.
- L'ALLIER JAMES, J. *Frame of Reference: NETg's Map to the Products, Their Structure and Core Beliefs*. PhD, 1997.
- LEE, J. y LEE, J. Context awareness by case-based reasoning in a music recommendation system. En *Ubiquitous Computing Systems*, vol. 4836, páginas 45–58. Springer, 2007. ISBN 978-3-540-76771-8.
- LEE, J., SUN, M. y LEBANON, G. A comparative study of collaborative filtering algorithms. *ArXiv Report 1205.3193*, 2012.
- LEHSTEN, P., ZENDER, R., LUCKE, U. y TAVANGARIAN, D. A service-oriented approach towards context-aware mobile learning management systems. En *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops*, páginas 268–273. IEEE Computer Society, 2010. ISBN 978-1-4244-6606-1.
- LEMIRE, D. y MACLACHLAN, A. Slope one predictors for online rating-based collaborative filtering. En *Proceedings of SIAM International Conference on Data Mining*, páginas 471–475. Society for Industrial and Applied Mathematics, 2005.
- LINDEN, G., HANKS, S. y LESH, N. Interactive assessment of user preference models: The automated travel assistant. En *Proceedings of the*

- 6th International Conference on User Modeling*, páginas 67–78. Springer, 1997. ISBN 9783211829066.
- LINDEN, G., SMITH, B. y YORK, J. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, vol. 7(1), páginas 76–80, 2003. ISSN 1089-7801.
- LOIZOU, A. y DASMAHAPATRA, S. Recommender systems for the semantic web. En *Proceedings of the European Conference on Artificial Intelligence Recommender Systems Workshop*, páginas 76–80. 2006.
- LOPS, P., GEMMIS, M. y SEMERARO, G. Content-based recommender systems: State of the art and trends. En *Recommender Systems Handbook*, páginas 73–105. Springer, 2011. ISBN 978-0-387-85819-7.
- LOW, Y., GONZALEZ, J., KYROLA, A., BICKSON, D., GUESTIN, C. y HELLERSTEIN, J. M. GraphLab: a new framework for parallel machine learning. En *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, páginas 340–349. AUAI Press, 2010. ISBN 978-0-9749039-6-5.
- MA, H., KING, I. y LYU, M. R. Effective missing data prediction for collaborative filtering. En *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, páginas 39–46. ACM, 2007. ISBN 978-1-59593-597-7.
- MAHMOUD, Q. H. Provisioning context-aware advertisements to wireless mobile users. En *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo*, páginas 669–672. IEEE Computer Society, 2006. ISBN 1-4244-0367-7.
- MAK, H., KOPRINSKA, I. y POON, J. INTIMATE: a Web-Based movie recommender using text categorization. En *Proceedings of the 2003 IEEE International Conference on Web Intelligence*, páginas 602–605. IEEE Computer Society, 2003. ISBN 0-7695-1932-6.
- MANOUSELIS, N., DRACHSLER, H., VUORIKARI, R., HUMMEL, H. y KOPER, R. Recommender systems in technology enhanced learning. En *Recommender Systems Handbook*, páginas 387–415. Springer, 2011a. ISBN 978-0-387-85820-3.
- MANOUSELIS, N., DRACHSLER, H., VUORIKARI, R., HUMMEL, H. y KOPER, R. Recommender systems in technology enhanced learning. En *Recommender Systems Handbook*, páginas 387–415. Springer, 2011b. ISBN 978-0-387-85820-3.
- MASTHOFF, J. Group modeling: Selecting a sequence of television items to suit a group of viewers. *User Modeling and User-Adapted Interaction*, vol. 14(1), páginas 37–85, 2004. ISSN 0924-1868.

- MCCARTHY, K., REILLY, J., SMYTH, B. y MCGINTY, L. Generating diverse compound critiques. *Artificial Intelligent Review*, vol. 24(3-4), páginas 339–357, 2005. ISSN 0269-2821.
- MCGINTY, L. y REILLY, J. On the evolution of critiquing recommenders. En *Recommender Systems Handbook*, páginas 419–453. Springer, 2011. ISBN 978-0-387-85819-7.
- MCGINTY, L. y SMYTH, B. Improving the performance of recommender systems that use critiquing. En *Intelligent Techniques for Web Personalization*, 3169, páginas 114–132. Springer, 2005. ISBN 978-3-540-29846-5, 978-3-540-31655-8.
- MCGREAL, R. A typology of learning object repositories. En *Handbook on Information Technologies for Education and Training*, páginas 5–28. Springer, 2008. ISBN 978-3-540-74155-8.
- MCLAUGHLIN, M. R. y HERLOCKER, J. L. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. En *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, páginas 329–336. ACM, 2004. ISBN 1-58113-881-4.
- MCNEE, S. M., RIEDL, J. y KONSTAN, J. A. Being accurate is not enough: how accuracy metrics have hurt recommender systems. En *Proceedings of Conference on Human Factors in Computing Systems*, páginas 1097–1101. ACM, 2006. ISBN 1-59593-298-4.
- MCSHERRY, D. Diversity-Conscious retrieval. En *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, páginas 219–233. Springer, 2002. ISBN 3-540-44109-3.
- MELVILLE, P., MOONEY, R. J. y NAGARAJAN, R. Content-boosted collaborative filtering for improved recommendations. En *Proceedings of the 18th National Conference on Artificial Intelligence*, páginas 187–192. American Association for Artificial Intelligence, 2002. ISBN 0-262-51129-0.
- MICHLÍK, P. y BIELIKOVÁ, M. Exercises recommending for limited time learning. *Procedia Computer Science*, vol. 1(2), páginas 2821–2828, 2010. ISSN 1877-0509.
- MOBASHER, B., JIN, X., ZHOU, Y., BERENDT, B., HOTHÖ, A., MLADENIC, D., VAN SOMEREN, M., SPILIOPOULOU, M. y STUMME, G. Semantically enhanced collaborative filtering on the web. En *Web Mining: From Web to Semantic Web*, vol. 3209, páginas 57–76. Springer, 2003. ISBN 978-3-540-23258-2.

- MOONEY, R. J. y ROY, L. Content-based book recommending using learning for text categorization. En *Proceedings of the 5th ACM Conference on Digital libraries*, páginas 195–204. ACM, 2000. ISBN 1-58113-231-X.
- MUSTO, C. Enhanced vector space models for content-based recommender systems. En *Proceedings of the 4th ACM Conference on Recommender Systems*, páginas 361–364. ACM, 2010. ISBN 978-1-60558-906-0.
- O'MAHONY, M. P. y SMYTH, B. A recommender system for on-line course enrolment: an initial study. En *Proceedings of the 2007 ACM Conference on Recommender Systems*, páginas 133–136. ACM, 2007. ISBN 978-1-59593-730-8.
- OWEN, S., ANIL, R., DUNNING, T. y FRIEDMAN, E. *Mahout in Action*. Manning Publications, 2011. ISBN 978-1-93518-268-9.
- PAZZANI, M. y BILLSUS, D. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, vol. 27(3), páginas 313–331, 1997. ISSN 0885-6125.
- PAZZANI, M. y BILLSUS, D. *Content-Based Recommendation Systems*, vol. 4321, páginas 325–341. Springer, 2007. ISBN 978-3-540-72078-2.
- PAZZANI, M. J. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, vol. 13(5-6), páginas 393–408, 1999. ISSN 0269-2821.
- PIEDRA, N., CHICAIZA, J., LOPEZ, J., TOVAR, E. y MARTINEZ, O. Finding OERs with social-semantic search. En *Proceedings of the IEEE Global Engineering Education Conference*, páginas 1195–1200. IEEE Computer Society, 2011. ISBN 978-1-61284-642-2.
- PIEDRA, N., CHICAIZA, J., LÓPEZ, J., MARTÍNEZ, O. y CARO, E. An approach for description of open educational resources based on semantic technologies. En *Proceedings of the IEEE Education Engineering*, páginas 1111–1119. IEEE Computer Society, 2010. ISBN 978-1-4244-6570-5.
- PREE, W. Meta patterns - a means for capturing the essentials of reusable object-oriented design. En *Proceedings of the 8th European Conference on Object-Oriented Programming*, páginas 150–162. Springer, 1994. ISBN 3-540-58202-9.
- PU, P. y CHEN, L. User-involved preference elicitation for product search and recommender systems. *AI Magazine*, vol. 29(4), páginas 93–103, 2009. ISSN 0738-4602.
- PU, P., CHEN, L. y KUMAR, P. Evaluating product search and recommender systems for e-commerce environments. *Electronic Commerce Research*, vol. 8(1-2), páginas 1–27, 2008. ISSN 1389-5753, 1572-9362.

- PU, P. y FALTINGS, B. Personalized navigation of heterogeneous product spaces using SmartClient. En *Proceedings of the 7th International Conference on Intelligent User Interfaces*, páginas 212–213. ACM, 2002. ISBN 1-58113-459-2.
- PU, P. y FALTINGS, B. Decision tradeoff using example-critiquing and constraint programming. *Constraints*, vol. 9(4), páginas 289–310, 2004. ISSN 1383-7133.
- PU, P., FALTINGS, B., CHEN, L., ZHANG, J. y VIAPPIANI, P. Usability guidelines for product recommenders based on example critiquing research. En *Recommender Systems Handbook*, páginas 511–545. Springer, 2011. ISBN 978-0-387-85819-7.
- PUERTA MELGUIZO, M. C., BOVES, L., DESHPANDE, A. y RAMOS, O. M. A proactive recommendation system for writing: helping without disrupting. En *Proceedings of the 14th European Conference on Cognitive Ergonomics: invent! explore!*, páginas 89–95. ACM, 2007. ISBN 978-1-84799-849-1.
- RANGANATHAN, A. y CAMPBELL, R. H. An infrastructure for context-awareness based on first order logic. *Personal Ubiquitous Computing*, vol. 7(6), páginas 353–364, 2003. ISSN 1617-4909.
- RECIO-GARCÍA, J. A., DÍAZ-AGUDO, B. y GONZÁLEZ-CALERO, P. A. Prototyping Recommender Systems in jCOLIBRI. En *Proceedings of the 2008 ACM Conference on Recommender Systems*, páginas 243–250. ACM, 2008. ISBN 978-1-60558-093-7.
- RECKER, M. y WALKER, A. Supporting “word of mouth” social networks through collaborative information filtering. *Journal of Interactive Learning Research*, vol. 14(1), páginas 79–98, 2003. ISSN 1093-023X.
- RICCI, F., ARSLAN, B., MIRZADEH, N. y VENTURINI, A. ITR: a case-based travel advisory system. En *Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, páginas 613–627. Springer, 2002. ISBN 3-540-44109-3.
- RICCI, F., ROKACH, L. y SHAPIRA, B. Introduction to recommender systems handbook. En *Recommender Systems Handbook*, páginas 1–35. Springer, 2011. ISBN 978-0-387-85819-7.
- ROCCHIO, J. Relevance feedback in information retrieval. En *The SMART Retrieval System*, páginas 313–323. Prentice-Hall, Inc., 1971.
- RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Promoting strong personalization in content-based recommendation systems of

- learning objects. En *Proceedings of the XI International Symposium on Computers in Education*. 2009a. ISBN 978-989-20-1774-7.
- RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Recommendation in repositories of learning objects: A proactive approach that exploits diversity and navigation-by-proposing. En *Proceedings of the 9th IEEE International Conference on Advanced Learning Technologies*, páginas 543–545. IEEE Computer Society, 2009b. ISBN 978-0-7695-3711-5.
- RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. User-adaptive recommendation techniques in repositories of learning objects: Combining long-term and short-term learning goals. En *Proceedings of the 4th European Conference on Technology Enhanced Learning: Learning in the Synergy of Multiple Disciplines*, páginas 645–650. Springer, 2009c. ISBN 978-3-642-04635-3.
- RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Personalización en recomendadores basados en contenido y su aplicación a repositorios de objetos de aprendizaje. *IEEE-RITA*, vol. 5(1), páginas 31–38, 2010. ISSN 1932-8540.
- RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Combining personalization and diversity in a case-based recommendation strategy for the learning domain. En *Proceedings of the International Council for Educational Media and International Symposium on Computers in Education Joint Conference*, páginas 409–419. 2011a. ISBN 978-972-789-347-8.
- RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. An experimental analysis of the behaviour of a personalized case-based recommendation strategy for the learning domain. En *Proceedings of the 19th International Conference on Computers in Education*, páginas 135–137. National Electronics and Computer Technology Center, 2011b. ISBN 978-616-12-0188-3.
- RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. Un framework para el desarrollo de recomendadores basados en contenido aplicados a objetos de aprendizaje. En *Actas del 2º Taller sobre Ingeniería del Software en eLearning*, páginas 147–161. Universidad Complutense de Madrid, Área de Ciencias Exactas y de la Naturaleza, 2011c. ISBN 978-84-694-7325-2.
- RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. A framework for rapid prototyping of knowledge-based recommender systems in the learning domain. *Journal of Research and Practice in Information Technology*, vol. 44(2), páginas 167–181, 2012a. ISSN 1443-458X.

- RUIZ-INIESTA, A., JIMÉNEZ-DÍAZ, G. y GÓMEZ-ALBARRÁN, M. A hybrid user-centered recommendation strategy applied to repositories of learning objects. *International Journal of Web Based Communities*, vol. 8(3), páginas 302–321, 2012b. ISSN 1741-8216.
- SAE-UENG, S., PINYAPONG, S., OGINO, A. y KATO, T. Personalized shopping assistance service at ubiquitous shop space. En *Proceedings of the 22nd International Conference on Advanced Information Networking and Applications*, páginas 838–843. IEEE Computer Society, 2008. ISBN 978-0-7695-3096-3.
- SALAKHUTDINOV, R. y MNIH, A. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems*, vol. 20, páginas 1257–1264, 2008. ISSN 1049-5258.
- SALAKHUTDINOV, R., MNIH, A. y HINTON, G. Restricted boltzmann machines for collaborative filtering. En *Proceedings of the 24th International Conference on Machine Learning*, páginas 791–798. ACM, 2007. ISBN 978-1-59593-793-3.
- SALAMÓ, M., REILLY, J., MCGINTY, L. y SMYTH, B. Knowledge discovery from user preferences in conversational recommendation. En *Knowledge Discovery in Databases*, vol. 3721, páginas 228–239. Springer, 2005. ISBN 978-3-540-29244-9.
- SALTON, G., WONG, A. y YANG, C. S. A vector space model for automatic indexing. *Communications of ACM*, vol. 18(11), páginas 613–620, 1975. ISSN 0001-0782.
- SANTOS, O. y BOTICARIO, J. Recommendation strategies for promoting e-learning performance factors for all. En *Proceedings of the 6th Workshop on Intelligent Techniques for Web Personalization & Recommender Systems in conjunction with the 23rd AAAI Conference on Artificial Intelligence*, páginas 89–98. AAAI, 2008a. ISBN 978-1-57735-374-4.
- SANTOS, O. y BOTICARIO, J. Users experience with a recommender system in an open source standard-based learning management system. En *HCI and Usability for Education and Work*, vol. 5298, páginas 185–204. Springer, 2008b. ISBN 978-3-540-89349-3.
- SANTOS, O. C. y BOTICARIO, J. G. Recommendations for providing dynamic inclusive learning. En *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies*, páginas 491–495. IEEE Computer Society, 2008c. ISBN 978-0-7695-3167-0.
- SARWAR, B., KARYPIS, G., KONSTAN, J. y RIEDL, J. Item-based collaborative filtering recommendation algorithms. En *Proceedings of the 10th*

- International Conference on World Wide Web*, páginas 285–295. ACM, 2001. ISBN 1-58113-348-0.
- SCHAFER, J. B., FRANKOWSKI, D., HERLOCKER, J. y SEN, S. Collaborative filtering recommender systems. En *The Adaptive Web*, páginas 291–324. Springer, 2007. ISBN 978-3-540-72078-2.
- SHELL, G. P. y BURNS, M. Merlot: A repository of e-learning objects for higher education. *e-Service Journal*, vol. 1(2), páginas 53–64, 2002. ISSN 1528-8234.
- SCHILIT, B., ADAMS, N. y WANT, R. Context-Aware computing applications. En *Proceedings of the 1st Workshop on Mobile Computing Systems and Applications*, páginas 85–90. IEEE Computer Society, 1994. ISBN 978-0-7695-3451-0.
- SCHMITT, S. *simVar*: a similarity-influenced question selection criterion for e-sales dialogs. *Artificial Intelligence Review*, vol. 18(3-4), páginas 195–221, 2002. ISSN 0269-2821.
- SETTEN, M., POKRAEV, S., KOOLWAAIJ, J., BRA, P. y NEJDL, W. Context-Aware recommendations in the mobile tourist application COMPASS. En *Adaptive Hypermedia and Adaptive Web-Based Systems*, vol. 3137, páginas 515–548. Springer, 2004. ISBN 978-3-540-22895-0.
- SHANI, G. y GUNAWARDANA, A. Evaluating recommendation systems. En *Recommender Systems Handbook*, páginas 257–297. Springer, 2011. ISBN 978-0-387-85820-3.
- SHARDANAND, U. y MAES, P. Social information filtering: algorithms for automating “word of mouth”. En *Proceedings of the Conference on Human Factors in Computing Systems*, páginas 210–217. ACM, 1995. ISBN 0-201-84705-1.
- SHEARIN, S. y LIEBERMAN, H. Intelligent profiling by example. En *Proceedings of the 6th International Conference on Intelligent User Interfaces*, páginas 145–151. ACM, 2001. ISBN 1-58113-325-1.
- SHELTON, B. E., DUFFIN, J., WANG, Y. y BALL, J. Linking open course wares and open education resources: creating an effective search and recommendation system. *Procedia Computer Science*, vol. 1(2), páginas 2865–2870, 2010. ISSN 1877-0509.
- SHETH, B. y MAES, P. Evolving agents for personalized information filtering. En *Proceedings of the 9th Conference on Artificial Intelligence for Applications*, páginas 345–352. IEEE Computer Society, 1993. ISBN 0-8186-3840-0.

- SHIMAZU, H. ExpertClerk: a conversational Case-Based reasoning tool for-Developing salesclerk agents in E-Commerce webshops. *Artificial Intelligence Review*, vol. 18(3), páginas 223–244, 2002. ISSN 0269-2821.
- SIEMER, J. y ANGELIDES, M. Towards an intelligent tutoring system architecture that supports remedial tutoring. *Artificial Intelligent Review*, vol. 12(6), páginas 469–511, 1998. ISSN 0269-2821.
- SMYTH, B. Case-Based recommendation. En *The Adaptive Web*, vol. 4321, páginas 342–376. Springer, 2007. ISBN 978-3-540-72078-2.
- SMYTH, B. y COTTER, P. Surfing the digital wave. En *Proceedings of the 3rd International Conference on Case-Based Reasoning and Development*, páginas 561–571. Springer, 1999. ISBN 3-540-66237-5.
- SMYTH, B. y COTTER, P. Personalized electronic program guides for digital TV. *AI Magazine*, vol. 22(2), páginas 89–98, 2001. ISSN 0738-4602.
- SMYTH, B. y KEANE, M. T. Adaptation-guided retrieval: questioning the similarity assumption in reasoning. *Artificial Intelligence*, vol. 102(2), páginas 249–293, 1998. ISSN 0004-3702.
- SMYTH, B. y MCCLAVE, P. Similarity vs. diversity. En *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development*, páginas 347–361. Springer, 2001. ISBN 3-540-42358-3.
- SOLINGEN, R. y BERGHOUT, E. *The Goal/Question/Metric Method*. McGraw-Hill Publishing Company, 1999. ISBN 007-709553-7.
- SOSNOVSKY, S. y DICHEVA, D. Ontological technologies for user modelling. *International Journal of Metadata, Semantics and Ontologies*, vol. 5(1), páginas 32–71, 2010. ISSN 1744-2621.
- SOSNOVSKY, S. y GAVRILOVA, T. Development of educational ontology for C-programming. *International Journal of Information Theories & Applications*, vol. 13(4), páginas 303–308, 2006. ISSN 1313-0463.
- STEFANER, M., DALLA VECCHIA, E., CONDOTTA, M., WOLPERS, M., SPECHT, M., APELT, S. y DUVAL, E. MACE-enriching architectural learning objects for experience multiplication. En *Creating New Learning Experiences on a Global Scale*, vol. 4753, páginas 322–336. Springer, 2007. ISBN 978-3-540-75194-6.
- STEGEMANN, S. K., FUNK, B. y SLODOS, T. A blackboard architecture for workflows. En *Proceedings of the CAiSE Forum at the 19th International Conference on Advanced Information Systems Engineering*, vol. 247, páginas 57–60. CEUR-WS, 2007. ISSN 1613-0073.

- STERN, H., KAISER, R., HOFMAIR, P., KRAKER, P. y LINDSTAEDT, S. N. Content recommendation in APOSDLE using the associative network. *Journal of Universal Computer Science*, vol. 16(16), páginas 2214–2231, 2010. ISSN 0948-6968.
- TANG, T. y MCCALLA, G. Smart recommendation for an evolving e-learning system: Architecture and experiment. *International Journal on E-Learning*, vol. 4(1), páginas 105–129, 2003. ISSN 1537-2456.
- TENG, D., CHEN, N.-S. y LEE, C.-H. Enhancing english reading comprehension by integrating direct access to digital materials and scaffolded questionings in paper prints. En *Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies*, páginas 244–248. IEEE Computer Society, 2011. ISBN 978-0-7695-4346-8.
- THOMPSON, C. A., GÖKER, M. H. y LANGLEY, P. A personalized system for conversational recommendations. *Journal of Artificial Intelligence Research*, vol. 21(1), páginas 393–428, 2004. ISSN 1076-9757.
- TSAI, K. H., CHIU, T. K., LEE, M. C. y WANG, T. I. A learning objects recommendation model based on the preference and ontological approaches. En *Proceedings of the 6th International Conference on Advanced Learning Technologies*, páginas 36–40. IEEE Computer Society, 2006. ISBN 0-7695-2632-2.
- TSATSOU, D., MENEMENIS, F., KOMPATSIARIS, I. y DAVIS, P. C. A semantic framework for personalized ad recommendation based on advanced textual analysis. En *Proceedings of the 3rd ACM Conference on Recommender Systems*, páginas 217–220. ACM, 2009. ISBN 978-1-60558-435-5.
- VENKATESH, V., MORRIS, M. G., DAVIS, G. B. y DAVIS, F. D. User acceptance of information technology: Toward a unified view. *MIS Quarterly*, vol. 27(3), páginas 425–478, 2003. ISSN 0276-7783.
- VERBERT, K., MANOUSELIS, N., OCHOA, X., WOLPERS, M., DRACHSLER, H., BOSNIC, I. y DUVAL, E. Context-aware recommender systems for learning: A survey and future challenges. *IEEE Transactions on Learning Technologies*, vol. 5(4), páginas 318–335, 2012. ISSN 1939-1382.
- VOLLRATH, I., WILKE, W. y BERGMANN, R. Case-based reasoning support for online catalog sales. *IEEE Internet Computing*, vol. 2(4), páginas 47–54, 1998. ISSN 1089-7801.
- WALKER, A., RECKER, M. M., LAWLESS, K. y WILEY, D. Collaborative information filtering: A review and an educational application. *International Journal of Artificial Intelligence in Education*, vol. 14(1), páginas 3–28, 2004. ISSN 1560-4292.

- WANG, J., DE VRIES, A. P. y REINDERS, M. J. T. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. En *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, páginas 501–508. ACM, 2006. ISBN 1-59593-369-7.
- WANG, J., DE VRIES, A. P. y REINDERS, M. J. T. Unified relevance models for rating prediction in collaborative filtering. *ACM Transactions on Information Systems*, vol. 26(3), páginas 1–42, 2008. ISSN 1046-8188.
- WANG, T. I., TSAI, K. H., LEE, M. C. y CHIU, T. K. Personalized learning objects recommendation based on the semantic-aware discovery and the learner preference pattern. *Educational Technology & Society*, vol. 10(3), páginas 84–105, 2007. ISSN 1436-4522.
- WELD, D. S., ANDERSON, C., DOMINGOS, P., ETZIONI, O., GAJOS, K., LAU, T. y WOLFMAN, S. Automatically personalizing user interfaces. En *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, páginas 1613–1619. Morgan Kaufmann Publishers Inc., 2003. ISBN 978-0-127-05661-6.
- WILEY, D. A. *Learning Object design and sequencing theory*. PhD, Brigham Young University, 2000.
- WOERNDL, W., SCHUELLER, C. y WOJTECH, R. A hybrid recommender system for context-aware recommendations of mobile applications. En *Proceedings of the 23rd IEEE International Conference on Data Engineering Workshop*, páginas 871–878. IEEE Computer Society, 2007. ISBN 978-1-4244-0831-3.
- XUE, G.-R., LIN, C., YANG, Q., XI, W., ZENG, H.-J., YU, Y. y CHEN, Z. Scalable collaborative filtering using cluster-based smoothing. En *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, páginas 114–121. ACM, 2005. ISBN 1-59593-034-5.
- YAU, J. y JOY, M. A context-aware and adaptive learning schedule framework for supporting learners' daily routines. En *Proceedings of the 2nd International Conference on Systems*, página 31. IEEE Computer Society, 2007. ISBN 0-7695-2807-4.
- YERGLER, N. R. Search and discovery: OER's open loop. En *Proceedings of Open Education Conference*, páginas 603–611. Universitat Oberta de Catalunya, 2010.
- YUDELSON, M. y BRUSILOVSKY, P. NavEx: Providing navigation support for adaptive browsing of annotated code examples. En *Proceedings of*

- the 2005 Conference on Artificial Intelligence in Education: Supporting Learning through Intelligent and Socially Informed Technology*, páginas 710–717. IOS Press, 2005. ISBN 1-58603-530-4.
- ZAÍANE, O. Building a recommender agent for e-learning systems. En *Proceedings of the International Conference on Computers in Education*, páginas 55–59. IEEE Computer Society, 2002. ISBN 0-7695-1509-6.
- ZANKER, M. y JESSENITSCHNIG, M. Case-studies on exploiting explicit customer requirements in recommender systems. *User Modeling and User-Adapted Interaction*, vol. 19(1-2), páginas 133–166, 2009a. ISSN 0924-1868.
- ZANKER, M. y JESSENITSCHNIG, M. Collaborative Feature-Combination recommender exploiting explicit and implicit user feedback. En *Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing*, páginas 49–56. IEEE Computer Society, 2009b. ISBN 978-0-7695-3755-9.
- ZANKER, M., JESSENITSCHNIG, M. y SCHMID, W. Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, vol. 15(4), páginas 574–595, 2010. ISSN 1383-7133.
- ZAPATA, A., MENENDEZ, V. H., PRIETO, M. E. y ROMERO, C. A hybrid recommender method for learning objects. *IJCA Proceedings on Design and Evaluation of Digital Content for Education*, vol. 1, páginas 1–7, 2011. ISSN 0975-8887.
- ZHANG, M. y HURLEY, N. Avoiding monotony: improving the diversity of recommendation lists. En *Proceedings of the 2008 ACM Conference on Recommender Systems*, páginas 123–130. ACM, 2008. ISBN 978-1-60558-093-7.
- ZHAO, S., ZHOU, M. X., YUAN, Q., ZHANG, X., ZHENG, W. y FU, R. Who is talking about what: social map-based recommendation for content-centric social websites. En *Proceedings of the 4th ACM Conference on Recommender Systems*, páginas 143–150. ACM, 2010. ISBN 978-1-60558-906-0.
- ZHAO, X., ANMA, F., NINOMIYA, T. y OKAMOTO, T. Personalized adaptive content system for Context-Aware mobile learning. *International Journal of Computer Science and Network Security*, vol. 8(8), páginas 153–161, 2008. ISSN 1738-7906.
- ZIEGLER, C., MCNEE, S. M., KONSTAN, J. A. y LAUSEN, G. Improving recommendation lists through topic diversification. En *Proceedings of the 14th International Conference on World Wide Web*, páginas 22–32. ACM, 2005. ISBN 1-59593-046-9.

Hoy empieza todo

